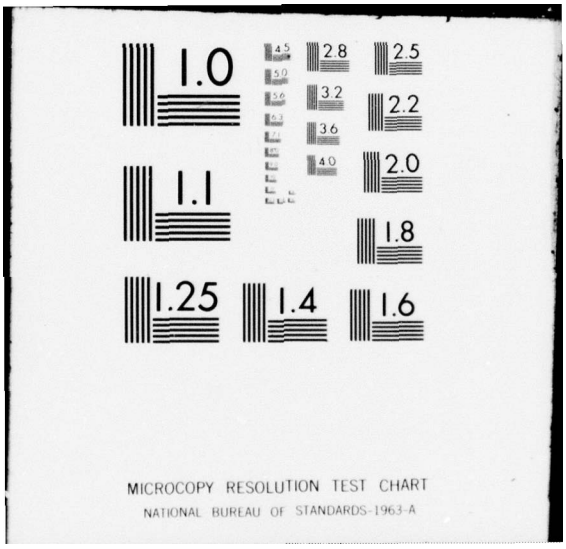
AD-A062 418 NAVAL RESEARCH LAB WASHINGTON D C F/G 20/9
QUEST: A QUASI-EQUILIBRIUM TRANSPORT CODE FOR HIGH-BETA SYSTEMS--ETC(U)
JUL 78 B H HUI, P C LIEWER, D L BOOK
UNCLASSIFIED NRL-MR-3813 SBIE-AD-E000 237 NL

END
DATE FILMED
3-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NRL Memorandum Report 3813

QUEST: A Quasi-Equilibrium Transport Code for High-Beta Systems

B. H. Hui

Science Applications, Inc.
McLean, Va. 22101

P. C. Liewer

Plasma Physics Division

and

D. L. Book

Laboratory for Computational Physics

July 1978

NAVAL RESEARCH LABORATORY
Washington, D.C.

(14) NRL-MR-3813

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NRL MEMORANDUM REPORT 3813
2. GOVT ACCESSION NO. (9) Memorandum rept.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): (6) QUEST: A QUASI-EQUILIBRIUM TRANSPORT CODE FOR HIGH-BETA SYSTEMS
5. TYPE OF REPORT & PERIOD COVERED: Interim report on a continuing NRL problem.
6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): (10) B. H. Hui, P. C. Liewer, D. L. Book
8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Research Laboratory, Washington, D. C. 20375
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NRL Problem No. 77H02-37; DOE Project EX-76-A-34-1006

11. CONTROLLING OFFICE NAME AND ADDRESS: (18) SBIE
12. REPORT DATE: (11) Jul[illegible]978
13. NUMBER OF PAGES: 69

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): (19) AD-E000 237
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This work was sponsored by the Department of Energy, Project No. EX-76-A-34-1006.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Liners
Field Reversal
Thermal Transport
Pressure Balance
beta

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the quasi-equilibrium compression and transport code, QUEST. The code is designed to study diffusion, thermal conduction, and ohmic and compressional heating in high-β plasma systems (β = ratio of plasma pressure to magnetic field pressure). These systems include Z-pinches, theta pinches, belt pinches and reversed field configurations. The code allows variations only in the radial direction and operates on a slow (diffusion and compression) time scale so that pressure balance is maintained. The conducting wall at the outer boundary of the system can be moved arbitrarily to simulate plasma compression produced by an imploding liner. This report also provides instructions on the use of the code, including a listing and sample input and output for verification.

beta

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

251950

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

QUEST: A Quasi-Equilibrium Transport Code for High-Beta Systems

I. INTRODUCTION

There are an increasing number of high-beta systems where the plasma dynamical time, defined as the time it takes the plasma to adjust to a force imbalance, is much shorter than the plasma lifetime. In reversed-field configurations, pinches and, perhaps, mirrors, the confinement time will be determined by diffusion processes, either classical or anomalous. Also, in liner experiments there is compressional heating on the slow (liner-speed) time scale. Diffusion and liner times are much longer than plasma dynamical times, which are typically the sonic and Alfven transit times. Thus to avoid prohibitively stringent time-step limitations it is necessary to develop a code which operates on the slower time scales.

It is well known that in low-β systems such as tokamaks, one can eliminate the dynamical time scale by neglecting the diffusion of the toroidal magnetic flux to lowest order and thus obtain an algebraic expression for the diffusion velocity and coefficient. However, in high-β ($\beta \simeq 1$) systems, the particle and magnetic flux diffusion are comparable and it is not possible to write an expresssion for the diffusion velocity and coefficients in closed from (see, e.g. Liewer and Davidson[1]).

But the diffusion processes are slow compared with the time for the plasma to reach force balance, so that the plasma essentially remains in equilibrium throughout the process. In a manner similar to that employed earlier to study impurity transport,[2] the QUEST code makes use of this to eliminate the dynamical time scale. Once diffusion has taken place so that the initial equilibrium plasma configuration moves slightly out of force balance, the code returns the

Manuscript submitted July 5, 1978.

configuration to equilibrium by assuming the plasma relaxes adiabatically and conservatively, i.e., with the particle number, magnetic flux and entropy held constant. The plasma is then allowed to diffuse again, and the relaxation repeated. This eliminates the need to solve the momentum equation, and thus eliminates the dynamical time scale. Thus it becomes feasible to model the evolution of the system over times as long as 10^3–10^4 sound or Alfven transit times, which would require as many as 10^6 time steps with conventional algorithm.

Compressional heating is included in the same way. After the plasma is compressed slightly by, say, inward motion of the outer wall, the plasma is again slightly out of force balance. Again, the configuration is returned to equilibrium assuming an adiabatic relaxation.

Thus far the code has been used to study the classical diffusion in the SEEBIE reversed magnetic field experiment at NRL.[3] This experiment has comparable axial and azimuthal magnetic fields. The code has also calculated successive states of an ideal lossless (i.e., with all transport processes turned off) reversed field configuration.

In the future, the code will be used to study strong compressional heating of such configurations both for the present parameters and for reactor parameters. Anomalous transport coefficients will be added. It is also planned to study the stability of such configurations by using the equilibrium configurations generated by QUEST as input to MHD and tearing mode stability codes.[4] (Thus QUEST can stand for quasi-equilibrium stability and transport.)

The organization of this report is as follows. In Section II, the equations solved by the code are presented, and the method of solution summarized. In Section III, we describe how to run the code. The input parameters and interpretation of the output are discussed. In Section IV, a description of each subroutine, its purpose and the solution methods used are described. Section V briefly describes the results obtained when measured SEEBIE profiles are used to initialize studies of adiabatic compression and classical transport separately. Appendix A contains

a description of the iteration method used to relax the grid to equilibrium. The code is listed in Appendix B. Sample input and output appears in Appendix C.

II. SUMMARY OF CODE

In this section, the equations solved by QUEST are presented and the method of solution described briefly.

A. Basic Equations

The evolution of a quasi-neutral plasma of two species [electrons (e) and ions (i)] is fully described by the equations

$$n = n_e = n_i, \tag{1}$$

$$nm_i \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = \frac{1}{c} \mathbf{J} \times \mathbf{B} - \nabla p; \tag{2}$$

$$\nabla \times \mathbf{B} = 4\pi \mathbf{J}/c; \tag{3}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -c \nabla \times \mathbf{E}; \tag{4}$$

$$p = nk_B(T_e + T_i); \tag{5}$$

$$\frac{3}{2} \frac{\partial}{\partial t} (nT_\alpha) + \frac{3}{2} \nabla \cdot (nT_\alpha \mathbf{v}) + nT_\alpha \nabla \cdot \mathbf{v} = Q_\alpha + \nabla \cdot (K_{\parallel}^{\alpha} \nabla_{\parallel} T_\alpha + K_{\perp}^{\alpha} \nabla_{\perp} T_\alpha), \quad \alpha = i, e; \tag{6}$$

$$\mathbf{E} + \frac{1}{c} \mathbf{v} \times \mathbf{B} = \eta_{\perp} \mathbf{J}_{\perp} + \eta_{\parallel} \mathbf{J}_{\parallel}, \tag{7}$$

where we have assumed a scalar pressure p, and Q_α is a heating term which includes resistive heating. The units are cgs through out this report and in the code except for the temperatures, which are in electron volts. Boltzmann's constant is $k_B = 1.6 \times 10^{-12}$ ergs/eV. The subscripts $\perp$ and $\parallel$ refer to directions perpendicular and parallel to the magnetic field **B**. The atomic number Z of the ions is taken to be unity.

We assume a cylindrically symmetric system which is infinite in the axial (z) direction. The equations then become, using Braginskii's[5] transport coefficients

$$\frac{\partial n}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}(nrv_r) = 0; \tag{8}$$

$$nm_i\left(\frac{\partial v_r}{\partial t} + v_r\frac{\partial v_r}{\partial r}\right) = \frac{1}{c}(J_\theta B_z - J_z B_\theta) - \frac{\partial p}{\partial r}; \tag{9}$$

$$\frac{3}{2}\frac{\partial}{\partial t}(nT_\alpha) + \frac{3}{2r}\frac{\partial}{\partial r}(nrv_r T_\alpha) + \frac{nT_\alpha}{r}\frac{\partial}{\partial r}(rv_r) = \tag{10}$$

$$Q_\alpha + \frac{1}{r}\frac{\partial}{\partial r}\left[rK^\alpha\frac{\partial T_\alpha}{\partial r}\right], \quad \alpha = i, e;$$

$$\frac{\partial B_\theta}{\partial t} + \frac{\partial}{\partial r}(v_r B_\theta) = \frac{\partial}{\partial r}\left[\frac{D_2}{r}\frac{\partial}{\partial r}(rB_\theta)\right] - \frac{\partial}{\partial r}\left[D_1\frac{\partial B_z}{\partial r}\right]; \tag{11}$$

$$\frac{\partial B_z}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}(rv_r B_z) = \frac{1}{r}\frac{\partial}{\partial r}\left[rD_3\frac{\partial B_z}{\partial r}\right] - \frac{1}{r}\frac{\partial}{\partial r}\left[D_1\frac{\partial}{\partial r}(rB_\theta)\right]. \tag{12}$$

Here

$$D_1 = \frac{c^2}{4\pi}(\eta_\parallel - \eta_\perp)\sin\xi\cos\xi; \tag{13}$$

$$D_2 = \frac{c^2}{4\pi}[\eta_\perp + (\eta_\parallel - \eta_\perp)\cos^2\xi]; \tag{14}$$

$$D_3 = \frac{c^2}{4\pi}[\eta_\perp + (\eta_\parallel - \eta_\perp)\sin^2\xi], \tag{15}$$

where $\cos\xi = B_z/B$, $\sin\xi = B_\theta/B$, and where

$$\eta_\perp = \frac{m_e}{ne^2\tau_e}; \tag{16}$$

$$\eta_\parallel = 0.51\,\eta_\perp; \tag{17}$$

$$\tau_e = 3.5\times 10^5 T_e^{3/2}/n\lambda; \tag{18}$$

$$\tau_i = 3.0\times 10^7 (m_i/2m_p)^{1/2} T_i^{3/2}/n\lambda; \tag{19}$$

$$\lambda = \begin{cases} 23.4 - 1.15\ln n + 3.45\ln T_e, & T_e < 50\text{eV}, \\ 25.3 - 1.15\ln n + 2.3\ln T_e, & T_e > 50\text{eV}; \end{cases} \tag{20}$$

$$K_e \equiv K_\perp^e = \frac{4.60nk_B T_e}{m_e\omega_{ce}^2\tau_e}; \tag{21}$$

$$K_i \equiv K_\perp^i = \frac{2nk_B T_i}{m_i\omega_{ci}^2\tau_i}; \tag{22}$$

$$\omega_{c\alpha} = \frac{eB}{m_\alpha c};$$

$$Q_i = \frac{3m_e}{m_i}\frac{n}{\tau_e}(T_e - T_i) + Q_{iA}; \tag{24}$$

$$Q_e = \frac{1}{k_B}\left(\eta_{\parallel} J_{\parallel}^2 + \eta_{\perp} J_{\perp}^2\right) - Q_i + Q_{eA} \tag{25}$$

$$= \frac{4\pi}{k_B c^2}\left[D_2 J_z^2 + D_3 J_\theta^2 + 2D_1 J_\theta J_z\right] - Q_i + Q_{eA}.$$

In obtaining the expression for the ion heat conductivity in Eq. (22), we have assumed the ions are magnetized, $\omega_{ci}\tau_i >> 1$. The $Q_{\alpha A}$ are any additional heating or cooling terms which may be included.

Equations (7-25) are the complete set of equations solved by the code. They are not, however, solved by a time integration, but rather in the quasiequilibrium approximation as described below.

In the absence of any diffusive processes, the right hand sides of Eqs. (6-7) vanish and Eq. (8) and Eqs. (10-12) can be written in the form

$$\frac{\partial}{\partial l} F + \frac{1}{r}\frac{\partial}{\partial r}\left(r v_r F\right) = 0 \tag{26}$$

where

$$F_1 = n \qquad \text{[Eq.(8)]};$$
$$F_2 = n^{1/3} T \qquad \text{[Eq.(10)]};$$
$$F_3 = B_\theta / r \qquad \text{[Eq.(11)]};$$
$$F_4 = B_z \qquad \text{[Eq.(12)]};$$

The four quantities $F_1 - F_4$ times the volume element $2\pi r dr$ are the usual conserved quantities: particle number, entropy, and θ- and z-projections of magnetic flux, respectively. Conservation of energy can be derived analytically from Eqs. (8-12) if the transport terms are dropped, but, in general, need not hold for numerical approximations that conserve the four quantities associated with $F_1 - F_4$.

B. Method of Solution

In the code Eqs. (7-25) are not integrated on the dynamical time scale. Instead, as discussed in the introduction, it is assumed that the processes which destroy the equilibrium [the

right hand sides of Eqs. (8-12)] operate on a time scale long compared to the time to bring the plasma to force balance. This allows Eqs. (7-25) to be solved in two steps. The plasma starts in equilibrium

$$\frac{1}{c}\,(J_\theta B_z - J_z B_\theta) = \frac{\partial p}{\partial r} \tag{27}$$

Thus, from Eq. (9) $v_r = 0$. Equations (7-25), with $v_r = 0$ (i.e., only the right hand sides of Eqs. (8-12) contribute to the time rates of change of the dependent variables) are then advanced using a fully time- and space-centered implicit scheme. The diffusion results in a force imbalance across the grid ($\mathbf{J} \times \mathbf{B} \neq \nabla p$). The configuration is brought back to equilibrium by an adiabatic relaxation of the grid: each cell has associated with it a value of the density n_j, an entropy function $T_{\alpha j}/n_j^{1/3}$, and values of $B_{\theta j}/r_j$ and B_{zj}. Each of these four quantities multiplied by the volume, $dV_j = 2\pi r_j \Delta r_j$ where Δr_j is the width of j^{th} cell, is held constant while an iteration on the cell widths Δr_j is performed until pressure balance is again restored. (The numerical procedure for doing this is described in Appendix A.) This is equivalent to finding the steady state solution $\left\{\frac{\partial v_r}{\partial t} = v_r = 0\right\}$ to Eq. (27) plus the momentum equation (9).

In addition, the code allows a third step, a compressional heating. The outer radius R of the cylinder moves according to

$$\frac{dR}{dt} = -V_{wall} \tag{28}$$

Thus the outer gridpoint is moved in a time step, Δt, by an amount

$$\Delta R = V_{wall}\Delta t \tag{29}$$

This also results in a force imbalance across the grid. The grid is again brought back to equilibrium by mean of the adiabatic relaxation described above.

The boundary conditions currently in the code specify $T_\perp = T_\parallel = 0$ at the axis and at the wall, so that the material pressure vanishes at both ends of the coordinate system. The

wall is taken to be a perfect thermal and electrical conductor and impervious to material flow, so that $v_r = V_{wall}$ there. The boundary conditions are used in initializing the code, solving the transport and pressure equilibration equations, and outputting the data. If different conditions are implemented in their place, changes should be made consistently throughout the code.

III. RUNNING THE CODE

In this Section, we discuss how the code is run. We describe first the input parameters, then the output. The code itself is treated here as a black box. Units are cgs throughout except for temperatures, which are in eV. The ion species is assumed to be deuterium.

A. Input data

Input is entered through two namelists: INPUT, which contains the parameters for the equilibrium conditions, and KNOB, which contains logical switches for various options in the code.

1. Namelist INPUT

GAM	—	the adiabatic constant γ. Currently this must be set to 5/3.
DRG	—	original grid spacing
ETA	—	parameter specifying initial ion and electron temperature ratio, $\eta = T_i/T_e$
BZ0	—	initial axial field at the wall
RRIGHT	—	initial position of the wall
RWALL	—	initial wall radius (should equal RRIGHT)
RMIN	—	radius of wall at peak compression
TMID	—	time of peak compression
T0	—	initial electron temperature

DT — time step (Since code is implicit, there is no stability requirement on DT. The user should select it for a good representation of the diffusional and compressional effects.)

BTO — parameter determining initial B_θ field at the wall (see Section IV B).

NSTOP — maximum number of time steps

NPRINT — print output and page plots every NPRINT steps

NPLOT — produce calcomp plots every NPLOT steps if NPLOT is a multiple of NPRINT

2. Namelist KNOB

LDSP — if true, dissipative processes are allowed (i.e., right hand sides of Eqs. (7-25) are finite); otherwise, only adiabatic compression

LWALL — if true, the outer wall is moved in according to $\frac{d}{dt} R_{wall} = - V_{wall} = - \frac{\text{RRIGHT} - \text{RMIN}}{\text{TMID}}$; if false wall remains stationary.

LPLOT — if true, generates page plots and calcomp plots

LBOOK — currently must be true; eventually will allow a reverse bias theta-pinch option

LTURB — currently must be false; eventually will allow for turbulent transport

The code is now set to simulate the experiment SEEBIE and the program set for 28 grid points. (This can be changed by a macro instruction.)

B. Output

Various grid quantities are printed out: R, the location of the grid points; B, the axial magnetic field (B_z); N, the particle density; TE, the electron temperature; TI, the ion temperature; PTOTAL = $nk_B(T_e + T_i)$; BP, the magnetic field pressure p_B, defined as

$$p_B = \frac{B^2}{8\pi} + \int_0^r dr' \, \frac{B_\theta^2(r')}{4\pi r'};$$

K, the total plasma plus magnetic field pressure in each cell (this must be the same in each cell or the grid relaxation has failed); BT, the azimuthal field B_θ; JZ, the axial current density; JT, the azimuthal current density; DELVZ, defined as $-J_z/ne$; DELV, defined as $-J_\theta/ne$; RESIST, the perpendicular resistivity $\eta_\perp$.

Also printed out are various extensive parameters characterizing the entire configuration:

NTOTAL — total number of particles

TOTE — total (plasma plus magnetic) energy

BENER — total magnetic energy

PENER — total plasma energy

ITOTZ — total axial current

ITOTT — total azimuthal current

BZTOTT — total axial flux

BTTOT — total azimuthal flux per cm

DIAM — diamagnetic signal Φ measured at $r = R(23)$ (set for SEEBIE),

$$\Phi = 2\pi \int_0^{R(23)} r \, dr \, B_z(r)$$

BTRAT — $B_\theta(23)/B_\theta^0(23)$ where B_θ^0 is the value at $t = 0$.

IV. DESCRIPTION OF CODE AND NUMERICAL METHODS USED

This section contains a description of each subroutine in the code. The purpose of each one is presented and the numerical methods used are described. The code itself is listed in Appendix B.

A. LINER — the main program

1. Initializes various constants such as species charge and mass. (Throughout code, species 1 is electrons and 2 is deuterium ions.)
2. Reads the input parameters through the namelists described in Section III A.
3. Calls the subroutine INCON which initializes the grid quantities and sets up an equilibrium configuration.
4. Calls the subroutines FLUX and LAGRAG to make sure the initial equilibrium configuration is a "finite-differenced" equilibrium.
5. Calls the subroutine STEP which now takes over control of the time evolution of the initial configuration.

B. Subroutine INCON

Function: initializing the grid quantities.

There are NRP grid points and NR = NRP-1 cells. Quantities can be defined on the grid point [e.g., RG(J) = 1/2 [R(J) + R(J + 1)]. Quantities ending in G (RG, DENG, etc.) are defined on the grid. Otherwise they are cell quantities (R, DEN, etc.). Most variable names are self explanatory. DR(J) is the array of cell sizes, DR(J) = RG(J + 1) - RG(J). The initial cell quantities are stored in arrays with the same names but with the suffix OLD, e.g., BZOLD(J) = BZ(J) at $t = 0$.

In this version, first the B_θ and B_z fields are calculated:

$$B_\theta(r) = \frac{\text{BTO}}{r} \int_0^r dr' \, r' e^{-(r' - \text{RJP})^2/\text{DRJZ}}$$

$$B_z(r) = -\text{BZOL} + \frac{(\text{BZO} + \text{BZOL})}{\pi \cdot \text{DRJT}} \int_0^r dr' \, e^{-(r' - \text{RJP})^2/\text{DRJT}}$$

where BTO, BZO, DRJZ, DRJT and RJP are input parameters. BZOL is calculated in INCON so that the plasma pressure is zero on axis. BZO is approximately the BZ field at the wall, RJP is the radius at which the current peaks, DRJZ and DRJT are related to the widths of the current profiles and BTO is related to the B_θ field at the wall.

Once the magnetic field profiles are calculated, the plasma pressure profile is calculated from force balance:

$$\frac{\partial}{\partial r}\left[p + \frac{B_\theta^2 + B_z^2}{8\pi} + \int_0^r \frac{dr'}{r'} \frac{B_\theta^2(r')}{4\pi}\right] = 0$$

The temperature profiles are specified and the density profiles then calculated from the pressure and temperature profiles.

C. Subroutine STEP

Function: controls the time loop. Successive actions are as follows:

1. Update NSTEP, the number of time steps taken and CLOCK, the time, and calculate the adiabatic invariants by calling the subroutine FLUX.
2. Move the outer wall in (if LWALL is true; see Section IV A) by calling the subroutine WALL.
3. Readjust the grid adiabatically to achieve force balance by calling the subroutine LAGRAG.
4. Calculate the new grid quantities from the adiabatic invariants and the new cell sizes. This is done in by calling the subroutine OUTPUT.

5. Allow diffusion and dissipation if LDSP is true and then returns the grid to force balance by calling the subroutine FLOW.

6. Produce output if time step condition is met by calling OUTPUT.

These six steps are then repeated NSTEP times.

D. Subroutine FLUX

Function: calculate the four adiabatic invariants [see Eq. (26)] which are held constant during the grid relaxation to achieve pressure balance. The four quantities actually calculated are

MASS (J) — MASS in the J*th* cell = $m_\alpha n(J) \times \pi(RG(J+1))^2 - (RG(J))^2]$

$S_\alpha(J)$ — entropy function, $S_\alpha(J) = P_\alpha(J)/[m_\alpha n(J)]^{5/3}$

BZFLX (J) — axial flux = $BZ(J) \times \pi\ [(RG(J+1))^2 - (RG(J))^2]$

BTFLX(J) -- azimuthal flux $BT(J) \times \pi[RG(J+1) - RG(J)]$

E. Subroutine WALL

Function: computes the wall radius RWALL as a function of time (denoted CLOCK) according to RWALL = RMIN + (RRIGHT − RMIN) × ABS (CLOCK − TMID) / TMID. The parameters are described in Section IV A.

F. Subroutine LAGRAG

Function: relax the grid to force balance by using an iteration process developed by Boris,[2] described in Appendix A. It calls the subroutine DDELR.

G. Subroutine FLOW

Function: Sets up the tridiagonal matrices to solve the equations for advancing T_e, T_i, B_θ, B_z [(eqs. (10-12) with $v_r = 0$)]. These equations are solved implicity by inverting the tridiagonal matrices. Furthermore, since the transport coefficients (K_α, $\eta_\perp$, etc.) depend on T_α and **B**, an iteration is performed on **B**, T_α and the transport coefficients until the desired accuracy is achieved. This is done with the following procedure at each time step:

1. Original values of **B**, T_α and the transport coefficients are stored (these matrices have names prefixed with an O, e.g., OBZ (J) and OKAPE(J).

2. An iteration loop is preformed to achieve the desired accuracy.

   a. The subroutine COLL is called to calculate the transport coefficient using the current (iterating) values of **B** and T_α.

   b. The tridiagonal matrtices are evaluated using transport coefficients of the form

$$K_\alpha = \frac{1}{2}\,(K_\alpha^n + K_\alpha^o)$$

   where K_α^n is the current transport coefficient [(calculated in step (2a)] and K_α^o is the original, stored value [(calculated in step (1)].

   c. The new values of T_α and **B** are calculated by inverting the tridiagonal matrices (by calling the subroutine TRID).

   d. The new temperature is compared to the previous value to see if the desired accuracy has been achieved.

H. Subroutine Coll

Function: Calculates the classical transport terms appearing in eqs. (10-12):

electron-ion collision frequency

TAU (I, 1) = τ_e, defined in Eq. (18)

ion-ion collision frequency

TAU (I,2) = τ_i, defined in Eq. (19)

Perpendicular and parallel resistivities

RESPRP (I) = $\eta_\perp$, defined in Eq. (16)

RESPAR(I) = $\eta_\parallel$, defined in Eqn. (17)

Magnetic field diffusion coefficient

BDONE(I) = D_1, defined in Eq. (13)

BD2(I) = D_2, defined in Eq. (14)

BD3(I) = D_3, defined in Eq. (15)

ion heating rate

HT (I, 2) = Q_i (with $Q_{iA} = 0$), defined in Eq. (24)

electron heating rate

HT(I, 1) = Q_e defined in Eq. (25) [In the Seebie run, there was a Q_{eA} which modeled thermal conduction losses out the end]

cross field thermal conductivity

KAPE(I) = K_e, defined in Eq. (21)

KAPI(I) = K_i, defined in Eq. (22)

V. EXAMPLES

A. Adiabatic Compression

We can turn off the transport effects and heat sources and sinks by setting $Q_\alpha = K_\alpha = \eta = 0$. The total energy is given by

$$W_{tot} = 2\pi \int_0^{R_{wall}} rdr\left[\frac{3}{2}\, nk_B(T_e + T_i) + \frac{B^2}{8\pi}\right]. \tag{30}$$

The rate of change of W_{tot} is now just equal to the work done on the system by the imploding liner,

$$\dot{W}_{tot} = 2\pi R_{wall} V_{wall} P, \tag{31}$$

where the total pressure P is given by

$$P = nk_B(T_e + T_i) + B^2/8\pi, \tag{32}$$

where the first term should be small or zero. The total pressure clearly increases with decreasing R_{wall}. There is an approximate power-law dependence which becomes exact in the trivial cases of pure magnetic flux or pure plasma[6].

For initial conditions, we chose the equilibrium profiles shown in Fig. 1. They were found by numerically fitting the profiles in the model used by Sethian, et. al.,[7] which were obtained in turn by an analytic fit to observations. There is, of course, no compression in this experiment, and the parameters of the observed configuration are unsuitable for a slow liner design (temperatures are of order 5-10 eV). For a test of the code and a preliminary exercise in preparation for realistic calculations using electron-beam-induced reversed-field configurations with scaled-up parameters, however, these initial conditions are a reasonable choice.

Using $V_{wall} = 10^4$ cm/sec in Eq. (28), we find the profiles shown in Fig. 2 after 0.3 msec and in Fig. 3 after 0.6 msec (i.e., after radial compression ratios of 1.75 and 7.0, respectively). The latter state has a maximum total pressure 915 times greater than in the initial state, and maximum B_z and T of 30.4 and 16.6 times those at $t = 0$. Note that the profiles have steepened considerably. This trend is expected to reduce the margin of MHD stability originally present, and (for sufficiently high compression) to destabilize the equilibrium. Even before this effect is noticeable, the steeper gradients are likely to drive microinstabilities which enhance particle and heat diffusion, thus tending to reduce the effectiveness of the compression in heating the plasma to thermonuclear conditions.

B. The SEEBIE Experiment

Returning to the basic system of equations (1-7), we adopt the Braginskii (classical) transport coefficients, Eqs. (13-25), and attempt to study the evolution of the initial profiles shown in Fig. 1, in the absence of compression. Two important competing effects are thereby introduced: generation of heat from decay of the magnetic fields (ohmic heating), and cross-field thermal transport, primarily carried by the ions.

In order to obtain agreement with the observed[7] decay of this configuration, it is necessary to include a model of axial heat losses by electron conduction. Though, strictly speaking, this requires a two-dimensional model, a crude representation is obtained by adding to Eq. (25) a term of the form

$$Q_{eA} = -2nT_e\bar{v}_e/l, \tag{33}$$

where $\bar{v}_e = (k_B T_e/m_e)^{1/2}$ is the electron thermal speed, and $l = 40$ cm. The resulting values of the diamagnetic signal Φ and of B_θ are compared with those measured by probes at approximately 6 cm radius. Although the oscillations observed experimentally in ΔB_z are not reproduced, the overall agreement is quite good, as is to be expected from a fluid description of this stable, cold plasma.

Appendix C contains the initial code parameters used to perform this calculations and that described in Section V A above, as well as the printed output (described in Section III B) initially and at the end of each run.

REFERENCES

1. P. C. Liewer and R. C. Davidson, Nucl. Fus. **17**, 85 (1977).

2. H. W. Bloomberg, J. P. Boris and B. Hui, NRL Memorandum Report No. 3155 (1975).

3. D. A. Hammer, A. E. Robson, K. A. Gerber and J. D. Sethian, Phys. Lett. **60**A, 31 (1977).

4. J. Kappraff, W. Grossmann, P. Rosenau, Paper I9, Controlled Thermonuclear Fusion Theory Meeting, San Diego, Calif., May 4-6, 1977; J. Kappraff and W. Grossmann (to be published).

5. S. I. Braginskii, *Reviews of Plasma Physics*, Vol. I (Consultants Bureau, New York, 1965), p. 205.

6. D. L. Book, D. A. Hammer and P. J. Turchi, Nucl. Fus. **18**, 159 (1978).

7. J. D. Sethian, D. A. Hammer, K. A. Gerber, D. N. Spector, A. E. Robson and G. C. Goldenbaum, NRL Memorandum Rept. No. 3648 (1977).

Appendix A

DETAILS OF THE PRESSURE BALANCE ALGORITHM

Here we show how the pressure balance Eq. (27) can be used to determine the motion of the numerical Lagrangian cells. We first take the reference position r_0 equal to zero and write Eq. (27) as

$$p(r) + \frac{B^2(r)}{8\pi} + \int_0^r dr\ \frac{B_\theta^2}{4\pi r} = K, \tag{A-1}$$

where

$$K = p(0) + \frac{B^2(0)}{8\pi}.$$

We write (A-1) in finite difference form

$$p(r_{i+1/2}) + \frac{B^2(r_{i+1/2})}{8\pi} + \frac{1}{4\pi} \sum_{k=1}^{i-1} \frac{B_\theta^2(r_{k+1/2})(r_{k+1} - r_k)}{(r_{k+1} + r_k)/2} + \frac{1}{4\pi} \frac{B_\theta^2(r_{i+1/2})(r_{i+1} - r_i)/2}{(r_{i+1} + r_i)/2} = K. \tag{A-2}$$

Pressure balance adjustment occurs under the condition that the magnetic flux, number of particles, and entropy, within each cell, are conserved. We now estimate p and B in (A-2) by reference to the conserved quantities. The magnetic fluxes are related to the fields by

$$B_\theta(r_{i+1/2}) = \frac{(\delta\Phi)_{i+1/2}}{r_{i+1} - r_i}, \tag{A-3}$$

$$B_z(r_{i+1/2}) = \frac{(\delta\Phi_z)_{i+1/2}}{\pi(r_{i+1}^2 - r_i^2)}. \tag{A-4}$$

To eliminate p from Eq. (A-1) we note that

$$p = \sum_\alpha S_\alpha \rho_\alpha^{\gamma_\alpha}, \tag{A-5}$$

where α refers to the species for which the entropy function S has been defined. When only one ion species is present (as in the versions of the code described here), we write $\rho_i = nm_i$ and $\rho_e = nm_e$. For multispecies problems we introduce the number density fraction of species α, $f_\alpha = n_\alpha/n$, whence

$$\rho_\alpha = \rho f_\alpha m_\alpha \Big/ \left(\sum_\alpha f_\alpha m_\alpha \right). \tag{A-6}$$

Now the total density in cell $i + 1/2$ is

$$\rho(r_{i+1/2}) = \frac{(\delta M)_{i+1/2}}{\pi(r_{i+1}^2 - r_i^2)}, \tag{A-7}$$

where $(\delta M)_{i+1/2}$ is the total mass in the cell. Substitution of (A-7) into (A-6) and the result into (A-5) allows us to write (A-2) in terms of the cell boundary positions and quantities held constant in the pressure balance adjustment:

$$\sum_\alpha S_\alpha(r_{i+1/2}) \left[\frac{(\delta M)_{i+1/2}}{\pi(r_{i+1}^2 - r_i^2)} \frac{f_\alpha m_\alpha}{\left(\sum_\beta f_\beta m_\beta \right)} \right]^{\gamma_\alpha} + \frac{(\delta\Phi_\theta)_{i+1/2}^2}{8\pi(r_{i+1} - r_i)^2} + \frac{(\delta\Phi_z)_{i+1/2}^2}{8\pi^3(r_{i+1}^2 - r_i^2)}$$
$$+ \frac{1}{2\pi} \sum_{k=1}^{i-1} \frac{(\delta\Phi_\theta)_{k+1/2}^2}{r_{k+1}^2 - r_k^2} + \frac{1}{4\pi} \frac{(\delta\Phi_\theta)_{i+1/2}^2}{r_{i+1}^2 - r_i^2} = K. \tag{A-8}$$

Consider now a system for which it is known that a pressure balance existed a short time previously. If, in the subsequent time interval the "constants" of the motion changed because of dissipative effects, then a new pressure balance with the "new" constants of the motion is required. Even neglecting dissipative effects a new pressure balance is required when a boundary moves or flux is added to the system.

The required pressure balance solution is obtained from (A-8) by first making a best guess for the new value of K. This is done by using the "new" values of the constants of the motion along with the cell positions computed at the previous time step. The K selected in this way will clearly depend on position, and we refer to the quantity as K_{i+1}. The true value of the cell boundary is, say, $r_i^{(0)} + \Delta_i^{(1)}$ where $r_i^{(0)}$ is our first guess or lower order iteration. If

$\Delta_i^{(1)} << r_i^{(0)}$ then an expansion r_i may be performed in (A-8), where only terms linear in the Δ's need be retained. The following recursion relation is then obtained:

$$\Delta_{i+1}^{(1)} = [K - K_{i+1}^{(0)} - \lambda_i^{(0)}(\Delta_i^{(1)}, \Delta_{i-1}^{(1)}, \cdots)]/A_{i+1}^{(0)}. \tag{A-9}$$

where $\lambda_i^{(0)}$ is composed of terms linear in the various Δ's and the A's are obtained straightforwardly by comparison with the expanded form of Eq. (A-8). If K is given, (A-9) may be solved for Δ_i, $i = 1, 2, \cdots$, successively because by symmetry we know that there can be no change in the positions of the first cell boundary , and $\Delta_1 = 0$. (Here we adopt the convention of beginning the numbering of the cell boundaries at the cylindrical axis.) Of course, this values of K will not give a proper solution unless the position of the last cell boundary, the physical wall, agrees with the value specified at that instant of time. For a stationary wall the corresponding $\Delta_w = 0$, while for a moving wall $\Delta_w = \beta_w \neq 0$. Hence we now construct two solutions of Eq. (A-9) corresponding to two values of K:

$$\Delta_{i+1}^{(a)} = [K^{(a)} - K_{i+1}^{(0)} - \lambda_i^{(0)}(\Delta_i^{(a)}, \Delta_{i-1}^{(a)}, \cdots)]/A_{i+1}^{(0)}, \tag{A-10a}$$

$$\Delta_{i+1}^{(b)} = [K^{(b)} - K_{i+1}^{(0)} - \lambda_i^{(0)}(\Delta_i^{(b)}, \Delta_{i-1}^{(b)}, \cdots)]/A_{i+1}^{(0)}. \tag{A-10b}$$

Because Eq. (A-9) is linear in K and Δ_i, the actual perturbed wall position can be formally expressed in terms of a linear combination of the solutions (A-10):

$$\beta_w = c^{(a)}\Delta_w^{(a)} + c^{(b)}\Delta_w^{(b)}, \tag{A-11}$$

where

$$c^{(a)} = 1 - \frac{\beta_w - \Delta_w^{(a)}}{\Delta_w^{(b)} - \Delta_w^{(a)}}, \tag{A-12a}$$

$$c^{(b)} = \frac{\beta_w - \Delta_w^{(a)}}{\Delta_w^{(b)} - \Delta_w^{(a)}} \tag{A-12b}$$

This determination of $c^{(a)}$ and $c^{(b)}$ now allows us to estimate the actual cell boundary positions, as well as K:

$$\Delta_{i+1}^{(1)} = c_i^{(a)}\Delta_{i+1}^{(a)} + c_i^{(b)}\Delta_{i+1}^{(b)}, \tag{A-13}$$

$$K = c_i^{(a)}K^{(a)} + c_i^{(b)}K^{(b)}. \tag{A-14}$$

The above expressions would be exact if the linearization used in the derivation of Eq. (9) were strictly valid and if the $K_{i+1}^{(0)}$ for all i were equal.

The values of Δ_{i+1} and K as determined from the above equations are treated as iterated quantities. Note that the iterated quantities do not depend on the choice $K^{(a)}$ and $K^{(b)}$. At this stage the best guess for the cell boundary positions is $r_i^{(1)} = r_i^{(0)} + \Delta_{i+1}$, where Δ_{i+1} is given by Eq. A(13). The iterated values at $r_i^{(1)}$ are use to find $K_{i+1}^{(1)}$, $\lambda_i^{(1)}$, and $A_{i+1}^{(1)}$, and the recursive equation for the second iteration is

$$\Delta_{i+1}^{(2)} = [K - K_{i+1}^{(1)} - \lambda_i^{(1)}(\Delta_i^{(2)}, \Delta_{i-1}^{(2)}, \cdots)]/A_{i+1}^{(1)}.$$

The procedure explained above is now repeated until the desired degree of convergence is obtained.

Appendix B

LISTING OF THE CODE, QUEST

The subroutines appear in alphabetical order. Member AAA1 is a macro which sets the number of grid points and species.

```
1* *MAC    VDIM
2*         PARAMETER NRP=14, NS=2, NR=NRP-1
3* *END    VDIM
```

```
      SUBROUTINE COLL(BDONE , BD2, BD3, FRAC, DELV)
C
C     COLL CALCULATES ALL COLLISIONAL TERMS AND SOLVE THE
C     E-EQUATION AND FIND VCOL
      IMPLICIT REAL*4 (A-Z)
      PARAMETER NRP=29, NS=2, NR=NRP-1
      COMMON/DUMP/ LDUMP
      COMMON/SWITCH/ LPRINT,LDSP,LWALL,LPLOT,LBOOK
      COMMON/COLPNT/ VIRG(NRP),ZETA(NRP), COIONE,COITWO,
     .  ZETONE,ZETTWO,COUNT
      COMMON/HANDLE/ LTURB
      COMMON/CON1/ PI, PI2, PI4, PI6, PI8, BK, C
      COMMON/CON2/ GAM,KNEW
      COMMON/SPECIE/ AM(NS), Q(NS)
      COMMON/GRID/ RG(NRP), R(NR), C1(NRP), C2(NRP), DR(NRP)
      COMMON/MTM/ BZ(NR), DEN(NR,NS), TEM(NR,NS), P(NR,NS), PTOT(NR)
     .  , BT(NRP)
      COMMON/OUT1/ TAUG(NRP,NS), HT(NRP,NS), QERU(NRP), QERT(NRP),
     .  SQEG(NRP), SQIG(NRP), DELVG(NRP), EG(NRP), WCG(NRP,NS)
     .  , VCOLG(NRP)
      COMMON/MTG/ BZG(NRP), DENG(NRP,NS), TEMG(NRP,NS), PG(NRP,NS),
     .  PTOTG(NRP), BTG(NRP)
      COMMON/SLOPE/ SDEN(NR,NS), STEM(NR,NS), SBZ(NR), E(NR)
      COMMON/KAPPA/ KAPE(NRP), KAPI(NRP)
      LOGICAL LTURB, LDUMP
      INTEGER I,J
      INTEGER COUNT,NSTEP
      LOGICAL LPRINT,LDSP,LWALL,LPLOT,LBOOK
      REAL*4 BETAG(NRP), ALFG(NRP), ALF(NRP), SBZG(NRP),
     .  DELV(NRP), RESIST(NRP),
     .  WC(NR,NS), TAU(NR,NS), SP(NR,NS),
     .  STEMG(NRP,NS), SPG(NRP,NS), BETA(NR)
     .  , VTHG(NRP,NS), RC(NRP,NS),
     .  VTH(NRP,NS), BD3(NRP), DELVZ(NRP),
     .   B(NRP), BG(NRP), BCONE(NRP), BD2(NRP)
     . ,GNUPRP(NRP),GNUPAR(NRP),RESPRP(NRP), RESPAR(NRP)
C
C       LOCAL BETA, WCE, COLL. TIME AT MID POINTS
C       THERMAL VELOCITIES, LARMOR RADIUS, MEAN FREE PATH
C
        FRAC1 = Q(2)/AM(1)/C
        FRAC2 = Q(2)/AM(2)/C
        DO 100 I=1,NR
       GNUPAR(I)=0.0
        GNUPRP(I)=0.0
        B(I) = SQRT(BZ(I)**2 + BT(I)**2)
        BETA(I) = DEN(I,1)*BK*(TEM(I,1) + TEM(I,2))/(B(I)**2/PI8)
        IF(TEM(I,1) .GT. 50.0) GO TO 2
        LAM = 23.4-1.15*ALOG10(DEN(I,1))+3.45*ALOG10(TEM(I,1))
        GO TO 3
 2      LAM = 25.3-1.15*ALOG10(DEN(I,1)) + 2.3*ALOG10(TEM(I,1))
```

```
*** MEMBER COLL

      3       CONTINUE                                                          0052000
              TAU(I,1) = 3.5E5*TEM(I,1)**1.5/(LAM*DEN(I,1))                     0053000
              TAU(I,2) = 3.0E7*TEM(I,2)**1.5/(LAM*DEN(I,1))                     0054000
              ALF(I) = AM(1)*DEN(I,1)/TAU(I,1)                                  0055000
              DO 110 J=1,NS                                                     0056000
              WC(I,J) = Q(2)*B(I)/AM(J)/C                                       0057000
              VTH(I,J) = SQRT(2.0*BK*TEM(I,J)/AM(J))                            0058000
              RC(I,J) = VTH(I,J)/ABS(WC(I,J))                                   0059000
    110       CONTINUE                                                          0060000
    100       CONTINUE                                                          0061000
C                                                                               0062000
C       LOCAL BETA, WCE,COLL. TIME AT GRID POINTS                               0063000
              DO 200 I=2,NR                                                     0064000
              BG(I) = SQRT(BZG(I)**2 + BTG(I)**2)                               0065000
              BETAG(I) = DENG(I,1)*BK*(TEMG(I,1) + TEMG(I,2))/                  0066000
                (BZG(I)**2/PI8)                                                 0067000
              IF(TEMG(I,1) .GT. 50.0) GO TO 4                                   0068000
              LAM = 23.4-1.15*ALOG10(DENG(I,1))+3.45*ALOG10(TEMG(I,1))          0069000
              GO TO 5                                                           0070000
    4         LAM = 25.3-1.15*ALOG10(DENG(I,1)) + 2.3*ALOG10(TEMG(I,1))         0071000
    5         CONTINUE                                                          0072000
              TAUG(I,1) = 3.5E5*TEMG(I,1)**1.5/(LAM*DENG(I,1))                  0073000
              TAUG(I,2) = 3.0E7*TEMG(I,2)**1.5/(LAM*DENG(I,1))                  0074000
              ALFG(I) = AM(1)*DENG(I,1)/TAUG(I,1)                               0075000
              DO 200 J=1,NS                                                     0076000
              WCG(I,J) = Q(2)*BG(I)/AM(J)/C                                     0077000
              VTHG(I,J) = SQRT(2.0*BK*TEMG(I,J)/AM(J))                          0078000
    200       CONTINUE                                                          0079000
              ALFG(1) = 0.0                                                     0080000
              ALFG(NRP) = 0.0                                                   0081000
C                                                                               0082000
C       GRADIENTS AT MID POINTS                                                 0083000
              DO 300 I=1,NR                                                     0084000
              SRZ(I) = (BZG(I+1) - BZG(I))/DR(I)                                0085000
              DO 310 J=1,NS                                                     0086000
              STEM(I,J) = BK*(TEMG(I+1,J) - TEMG(I,J))/DR(I)                    0087000
              SDEN(I,J) = (DENG(I+1,J) - DENG(I,J))/DR(I)                       0088000
    310       SP(I,J) = (PG(I+1,J) - PG(I,J))/DR(I)                             0089000
    300       CONTINUE                                                          0090000
C                                                                               0091000
C       GRADIENTS AT GRID POINTS                                                0092000
              FOS = 5.0/6.0                                                     0093000
              SRZG(1) = 0.0                                                     0094000
              DO 410 J=1,NS                                                     0095000
              STEMG(1,J) =0.0                                                   0096000
    410       SPG(1,J) =0.0                                                     0097000
              DO 430 I=2,NR                                                     0098000
              SRZG(I) = SRZ(I-1)*C1(I) + SRZ(I)*C2(I)                           0099000
              DO 420 J=1,NS                                                     0100000
              STEMG(I,J) = STEM(I-1,J)*C1(I) + STEM(I,J)*C2(I)                  0101000
              SPG(I,J) = SP(I-1,J)*C1(I) + SP(I,J)*C2(I)                        0102000
```

```
*** MEMBER CALL

      103*  420     CONTINUE                                                  0103000
      104*  430     CONTINUE                                                  0104000
      105*          SBZG(NRP) = 0.0                                           0105000
      106*          DO 431 J=1,NS                                             0106000
      107*          STEMG(NPP,J) = 0.0                                        0107000
      108*          SPG(NRP,J) = 0.0                                          0108000
      109*  431     CONTINUE                                                  0109000
      110* C                                                                  0110000
      111* C     CALL FOR ANOMALOUS TRANSPORT                                0111000
      112*        IF( .NOT. LTURB) GO TO 215                                  0112000
      113*           CALL TURB( DELVG,DELVZ,KCG,GNUPRP,GNUPAR,BG)             0113000
      114*  215     CONTINUE                                                  0114000
      115* C     CALCULATE B-DIFFUSION COEFFICIENTS                          0115000
      116*          MOE2 = AM(1)/Q(1)**2                                      0116000
      117*          CP = C**2/PI4                                             0117000
      118*          DO 370 I=2,NR                                             0118000
      119*           RESPRP(I)=MOE2/DENG(I,1)*(1./TAUG(I,1) + GNUPRP(I) )     0119000
      120*           RESPAR(I)=MOE2/DENG(I,1)*(.51/TAUG(I,1)+GNUPAR(I))       0120000
      121*          SSIN = BTG(I)/BG(I)                                       0121000
      122*          CCOS = BZG(I)/BG(I)                                       0122000
      123*           BDONE(I)=CP*(RESPAR(I)-RESPRP(I))*SSIN*CCOS              0123000
      124*           BD2(I)=CP*(RESPRP(I)*SSIN**2 +RESPAR(I)*CCOS**2)         0124000
      125*           BD3(I)=CP*(RESPRP(I)*CCOS**2+RESPAR(I)*SSIN**2)          0125000
      126*  370     CONTINUE                                                  0126000
      127*          BD2(1) = 0.0                                              0127000
      128*          BDONE(1) = 0.0                                            0128000
      129*          BD3(1) = 0.0                                              0129000
      130*          BDONE(NRP) = 0.0                                          0130000
      131*          BD2(NRP) = 0.0                                            0131000
      132*          BD3(NRP) = 0.0                                            0132000
      133* C                                                                  0133000
      134* C                                                                  0134000
      135* C     CALCULATE DEL-V-THETA, DEL-VZ, VCOLL AND THERMAL FORCE       0135000
      136*          DO 800 I=2,NR                                             0136000
      137*          DELVG(I) = -C*SBZG(I)/(PI4*Q(2)*DENG(I,2))                0137000
      138*  800     CONTINUE                                                  0138000
      139*          DELVG(NRP) = 0.0                                          0139000
      140*          DELVG(1) = 0.0                                            0140000
      141*          DO 811 I=1,NR                                             0141000
      142*          DELV(I) = (DELVG(I) + DELVG(I+1))/2.0                     0142000
      143*          DELVZ(I) = C/PI4*(RG(I+1)*BTG(I+1) - RG(I)*BTG(I))/       0143000
      144*     .      (P(I)*DR(I)*DEN(I,1)*Q(2))                             0144000
      145*  811     CONTINUE                                                  0145000
      146* C                                                                  0146000
      147* C     CALCULATE HEAT TERMS                                        0147000
      148*          RTMASS = AM(1)/AM(2)                                      0148000
      149*          DO 911 I=1,NR                                             0149000
      150*          HT(I,2) = 3.0*RTMASS*DEN(I,1)/TAU(I,1)*(TEM(I,1)-TEM(I,2))*BK  0150000
      151*           HT(I,1) = 0.5*(DEN(I,1)*Q(1))**2/CP* (                   0151000
      152*     .     (BD2(I) +BD2(I+1))*DELVZ(I)**2+(BD3(I)+BD3(I+1))*DELV(I)**2  0152000
      153*     .     +2.0*(BDONE(I)+BDONE(I+1))*DELVZ(I)*DELV(I) ) - HT(I,2)  0153000
```

```
*** MEMBER COLL

      154*  911     CONTINUE                                                  0150000
      155*          DO 912 I=1,NR                                             0155000
      156*          DELTE = HT(I,1)*FRAC/(DEN(I,1)*BK)                        0156000
      157*          DELTI = HT(I,2)*FRAC/(DEN(I,2)*BK)                        0157000
      158*          IF(ABS(DELTE) .LT. TEM(I,1)*0.2) GO TO 913                0158000
      159*          HT(I,1) = TEM(I,1)*0.2*DEN(I,1)*BK/FRAC                   0159000
      160*  913     CONTINUE                                                  0160000
      161*          IF(ABS(DELTI) .LT. TEM(I,2)*0.2) GO TO 912                0161000
      162*          HT(I,2) = TEM(I,2)*0.2*DEN(I,2)*BK/FRAC                   0162000
      163*  912     CONTINUE                                                  0163000
      164*          DO 910 I=2,NR                                             0164000
      165*          KAPE(I) = 4.66*DENG(I,1)*TEMG(I,1)*BK/(AM(1)*             0165000
      166*     .      WCG(I,1)**2*TAUG(I,1))                                 0166000
      167*          KAPI(I) = 2.0*DENG(I,2)*TEMG(I,2)*BK/(AM(2)*              0167000
      168*     .      WCG(I,2)**2*TAUG(I,2))                                 0168000
      169*  910     CONTINUE                                                  0169000
      170*          KAPE(1) = KAPE(2)                                         0170000
      171*          KAPI(1) = KAPI(2)                                         0171000
      172*          HT(NRP,1) = 0.0                                           0172000
      173*          HT(NRP,2) = 0.0                                           0173000
      174*          KAPE(NRP) = KAPE(NR)                                      0174000
      175*          KAPI(NRP) = KAPI(NR)                                      0175000
      176* C                                                                 0176000
      177*   1      CONTINUE                                                  0177000
      178*        RETURN                                                      0178000
      179*        END                                                         0179000
```

```
*** MEMBER DDELR

      1*       SUBROUTINE DDELR (TESTK, KOLD, DELR)
      2* C
      3* C     DDELR COMPUTES GRIDS DISPLACEMENTS DUE TO PRESSURE ADJUSTMENT
      4* C     TESTK = TRIAL TOTAL PRESSURE CONSTANT
      5* C     DELR = GRIDS DISPLACEMENTS
      6*          IMPLICIT REAL*4 (A-Z)
      7* *        VDIM
      8*          COMMON/CON2/ GAM, KNEW
      9*          COMMON/CON1/ PI, PI2, PI4, PI6, PI8, BK, C
     10*          COMMON/GRID/ RG(NRP), R(NR), C1(NRP), C2(NRP), DR(NRP)
     11*          COMMON/FLUXQ/ MASS(NRP,NS), S(NRP,NS), BZFLX(NRP),BTFLX(NRP)
     12*          REAL*4 DELR(NRP)
     13*          INTEGER I,J, K, L
     14* C
     15*          DELR(1) = 0.0
     16*          DO 100 I=1,NR
     17*          RSDIF = RG(I+1)**2 - RG(I)**2
     18*          PTERM = 0.0
     19*          DO 200 J=1,NS
     20*          PTERM = PTERM +  S(I,J)*(MASS(I,J)/(PI*RSDIF))**GAM
     21*  200     CONTINUE
     22*          BZTERM = BZFLX(I)**2/((PI*RSDIF)**2*PI8)
     23*          BTTERM = BTFLX(I)**2/(PI8*DR(I)**2)
     24*          BST = 0.0
     25*          BSTR = 0.0
     26*          IF(I .EQ. 1) GO TO 300
     27*          K= I -1
     28*          DO 400 L=1,K
     29*          RSD = RG(L+1)**2 - RG(L)**2
     30*          BST = BST + BTFLX(L)**2/(PI2*RSD)
     31*  400     BSTR = BSTR + BTFLX(L)**2/(PI*RSD**2)*(RG(L+1)*DELR(L+1)
     32*        .   - RG(L)*DELR(L))
     33*  300     BTH = BTFLX(I)**2/(PI4*RSDIF)
     34*          KOLD = PTERM + BZTERM + BTTERM + BST + BTH
     35*          ONE = (2.0*PTERM*GAM/RSDIF + BZTERM*4.0/RSDIF + BTTERM
     36*        .   *2.0/DR(I) + BTH*2.0/RSDIF)*RG(I+1)
     37*          TWO = ONE*RG(I)/RG(I+1)
     38*          DELR(I+1) = (KOLD - TESTK - BSTR + TWO*DELR(I))/ONE
     39*  100     CONTINUE
     40*          RETURN
     41*          END
```

```
*** MEMBER FLOW

      1*        SUBROUTINE FLOW (DT, TFACT, FRAC, NSTEP)                                0001000
      2* C                                                                             0002000
      3* C     FLOW ADVANCES DEN, TEMP AND B IN ONE TIME STEP DUE TO                   0003000
      4* C     COLLISIONS USING FCT                                                    0004000
      5* C                                                                             0005000
      6*           IMPLICIT REAL*4 (A -Z)                                             0006000
      7*           PARAMETER NRP=29, NS=2, NR=NRP-1                                   0007000
      8*           COMMON/SWITCH/ LPRINT,LDSP,LWALL,LPLOT,LBOOK                       0008000
      9*           COMMON/BC/ NFRAC, TFRAC, TLBASE, TRBASE                            0009000
     10*           COMMON/SPECIE/ AM(NS), Q(NS)                                       0010000
     11*           COMMON/TWO/ TOTE, VOL                                              0011000
     12*           COMMON/LOOP/ K, NSTP                                               0012000
     13*           COMMON/KAPPA/ KAPE(NRP), KAPI(NRP)                                 0013000
     14*           COMMON/FLUXQ/ MASS(NRP,NS), S(NRP,NS), BZFLX(NRP),BTFLX(NR)        0014000
     15*           COMMON/DUMP/ LDUMP                                                 0015000
     16*           COMMON/CON1/ PI, PI2, PI4, PI6, PI8, BK, C                         0016000
     17*           COMMON/CON2/GAM, KNEW                                              0017000
     18*           COMMON/MTG/ BZG(NRP), DENG(NRP,NS), TEMG(NRP,NS), PG(NRP,NS),      0018000
     19*          .          PTOTG(NRP), BTG(NRP)                                     0019000
     20*           COMMON/MTM/ BZ(NR), DEN(NR,NS), TEM(NR,NS), P(NR,NS), PTOT(NR)     0020000
     21*          .         , BT(NR)                                                  0021000
     22*           COMMON/GRID/ PG(NRP), R(NR), C1(NRP), C2(NRP), DR(NRP)             0022000
     23*           COMMON/OUT1/ TAUG(NRP,NS), HT(NRP,NS), QERU(NRP), QERT(NRP),       0023000
     24*          .   SQEG(NRP), SQIG(NRP), DELVG(NRP), EG(NRP), WCG(NRP,NS)         0024000
     25*          .    , VCOLG(NRP)                                                  0025000
     26*           REAL*4 RHO(NR), PRE(NR), PRI(NR),  RHON(NR),                       0026000
     27*          .   PREN(NR), PRIN(NR),  CE2(NR), CI2(NR), DE2(NR),                0027000
     28*          .   DE3(NR), DI3(NR), V1R(NR), BZO(NR), SP(NR), DELV(NR)           0028000
     29*          . , BDIF(NRP),CONE(NR), CTWO(NR), AA(NR), BB(NR), CC(NR),          0029000
     30*          .   DD(NR), DTEMI(NRP), DTEME(NRP), DBZ(NRP), DBT(NRP),            0030000
     31*          .   TTEME(NRP), TTEMI(NRP), TBZ(NRP), TBT(NRP), BDONE(NRP),        0031000
     32*          .   BD2(NRP), BD3(NRP), OBT(NRP), OBDONE(NRP), OBD2(NRP),          0032000
     33*          .   OBD3(NRP), OBDIF(NRP), OKAPI(NRP), OKAPE(NRP),OTEM(NR,NS),     0033000
     34*          .   OBZ(NR), FONE(NR), FTWO(NR), FREE(NR), FOUR(NR)                0034000
     35*          .   , RONE(NRP), R2(NRP), FACT(NR)                                 0035000
     36*           INTEGER NRM                                                        0036000
     37*           INTEGER I, J, K, L, NSTEP,NSTP, M, MM, N                           0037000
     38*           LOGICAL HALF, LDUMP, LPASS                                         0038000
     39*           LOGICAL LPRINT,LDSP,LWALL,LPLOT,LBOOK                              0039000
     40* C                                                                           0040000
     41*           NRM = NR - 1                                                       0041000
     42*           SDT=DT                                                             0042000
     43*           TIME = 0.0                                                         0043000
     44*   8       CONTINUE                                                           0044000
     45*           NSTP = 0                                                           0045000
     46*           DO 7 I=1,NR                                                        0046000
     47*           OBT(I) = BT(I)                                                     0047000
     48*           OBZ(I) = BZ(I)                                                     0048000
     49*           DO 7 J=1,NS                                                        0049000
     50*           OTEM(I,J) = TEM(I,J)                                               0050000
     51*   7       CONTINUE                                                           0051000
```

```
*** MEMBER FLOW

      52*  1        CONTINUE                                                        0052000
      53*           NSTP = NSTP + 1                                                 0053000
      54*           CALL COLL(BDONE, BD2, BD3, SDT, DELV)                           0054000
      55*           IF(NSTP .NE. 1) GO TO 3                                         0055000
      56*           DO 403 I=1,NRP                                                  0056000
      57*           DTEMI(I) = 0.0                                                  0057000
      58*           DTEME(I) = 0.0                                                  0058000
      59*           DBZ(I) = 0.0                                                    0059000
      60*           DBT(I) = 0.0                                                    0060000
      61*           OBDONE(I) = BDONE(I)                                            0061000
      62*           OBD2(I) = BD2(I)                                                0062000
      63*           OBD3(I) = BD3(I)                                                0063000
      64*           OKAPI(I) = KAPI(I)                                              0064000
      65*  403      OKAPE(I) = KAPE(I)                                              0065000
      66*  3        CONTINUE                                                        0066000
      67*           DO 404 I=1,NRP                                                  0067000
      68*           BDONE(I) = (BDONE(I) + OBDONE(I))/2.0                           0068000
      69*           BD2(I) = (BD2(I) + OBD2(I))/2.0                                 0069000
      70*           BD3(I) = (BD3(I) + OBD3(I))/2.0                                 0070000
      71*           KAPI(I) = (KAPI(I) + OKAPI(I))/2.0                              0071000
      72*  404      KAPE(I) = (KAPE(I) + OKAPE(I))/2.0                              0072000
      73* C                                                                        0073000
      74* C     ADVANCE TI                                                         0074000
      75*           DO 200 I=2,NRM                                                  0075000
      76*           DELRP = R(I+1) - R(I)                                           0076000
      77*           DELRM = R(I) - R(I-1)                                           0077000
      78*           CONE(I) = SDT*RG(I+1)/(R(I)*DR(I)*DELRP)                        0078000
      79*           CTWO(I) = SDT*RG(I)/(R(I)*DR(I)*DELRM)                          0079000
      80*           IONE = CONE(I)*KAPI(I+1)/(3.0*DEN(I,2))                         0080000
      81*           ITWO = CTWO(I)*KAPI(I)/(3.0*DEN(I,2))                           0081000
      82*           AA(I) =-ITWO                                                    0082000
      83*           BB(I) = 1.0 + IONE + ITWO                                       0083000
      84*           CC(I) =-IONE                                                    0084000
      85*           DD(I) =2.0*SDT*HT(I,2)/(3.0* DEN(I,1)*BK) + ITWO* OTEM(I-1,2)   0085000
      86*          .     + (1.0 - IONE - ITWO)* OTEM(I,2) + IONE* OTEM(I+1,2)       0086000
      87*  200      CONTINUE                                                        0087000
      88* C                                                                        0088000
      89* C     B.C. FOR TI AT R=0                                                 0089000
      90*           DELRP = R(2) - R(1)                                             0090000
      91*           CONE(1) = SDT*RG(2)/(R(1)*DR(1)*DELRP)                          0091000
      92*           IONE = CONE(1)*KAPI(2)/(3.0*DEN(1,2))                           0092000
      93*           ITWO = 2.0*KAPI(1)*SDT/(3.0*DEN(1,1)*DP(1)**2)                  0093000
      94*           BB(1) = 1.0 + IONE +ITWO                                        0094000
      95*           CC(1) = -IONE                                                   0095000
      96*           DD(1) =OTEM(1,2)*(1.0-ITWO) + 2.0*SDT*HT(1,2)/(3.0*DEN(1,2)*BK) 0096000
      97*          .     +IONE*(OTEM(2,2) - OTEM(1,2))                            0097000
      98* C                                                                        0098000
      99* C     B.C. TI AT WALL                                                    0099000
     100*           DELRM = R(NR) - R(NR-1)                                         0100000
     101*           CTWO(NR) = SDT*RG(NR)/(R(NR)*DR(NR)*DELRM)                      0101000
     102*           CONE(NR) = SDT*RG(NRP)/(R(NR)*DP(NR)**2)                        0102000
```

```
*** MEMBER FLOW

103*            IONE = CONE(NR)*2.0*KAPI(NRP)/(3.0*DEN(NR,2))                    0103000
104*            ITWO = CTWO(NR)*KAPI(NRP)/(3.0*DEN(NR,2))                        0104000
105*            AA(NR) = -ITWO                                                   0105000
106*            BB(NR) = 1.0 + ITWO +IONE                                        0106000
107*            CC(NR) = 0.0                                                     0107000
108*            DD(NR) = 2.0*SDT*HT(NR,2)/(3.0*DEN(NR,2)*BK) +                   0108000
109*           .   ITWO*OTEM(NR-1,2) + (1.0  -IONE-ITWO)*OTEM(NR,2)              0109000
110*            CALL TRID(TEM(1,2),AA, BB, CC, DD)                               0110000
111* C                                                                           0111000
112* C      ADVANCE TE                                                           0112000
113*            DO 50 I=2,NRM                                                    0113000
114*            EONE = CONE(I)*KAPE(I+1)/(3.0*DEN(I,1))                          0114000
115*            ETWO = CTWO(I)*KAPE(I)  /(3.0* DEN(I,1))                         0115000
116*            AA(I) =-ETWO                                                     0116000
117*            BB(I) = 1.0 + EONE+ ETWO                                         0117000
118*            CC(I) =-EONE                                                     0118000
119*            DD(I) =2.0*SDT*HT(I,1)/(3.0* DEN(I,1)*BK) + ETWO* OTEM(I-1,1)    0119000
120*           .   + (1.0 - EONE - ETWO)* OTEM(I,1) + EONE* OTEM(I+1,1)          0120000
121*  50       CONTINUE                                                         0121000
122* C                                                                           0122000
123* C      B.C. FOR TE AT R=0                                                   0123000
124*            EONE = CONE(1)*KAPE(2)/(3.0*DEN(1,1))                            0124000
125*            ETWO = 2.0*KAPE(1)*SDT/(3.0*DEN(1,1)*DR(1)**2)                   0125000
126*            BB(1) = 1.0 + EONE + ETWO                                        0126000
127*            CC(1) = -EONE                                                    0127000
128*            DD(1) =OTEM(1,1)*(1.0-ETWO) + 2.0*SDT*HT(1,1)/(3.0*DEN(1,1)      0128000
129*           .   *BK) + EONE*(OTEM(2,1) - OTEM(1,1))                          0129000
130* C                                                                           0130000
131* C      B.C. FOR TE AT WALL                                                  0131000
132*            ETWO = CTWO(NR)*KAPE(NR)/(3.0*DEN(NR,1))                         0132000
133*            EONE = CONE(NR)*2.0*KAPE(NRP)/(3.0*DEN(NR,1))                    0133000
134*            AA(NR) = -ETWO                                                   0134000
135*            BB(NR) = 1.0 + ETWO + EONE                                       0135000
136*            CC(NR) = 0.0                                                     0136000
137*            DD(NR) = 2.0*SDT*HT(NR,1)/(3.0*DEN(NR,1)*BK) +                   0137000
138*           .    ETWO*OTEM(NR-1,1) + (1.0 -EONE-ETWO)*OTEM(NR,1)             0138000
139*            CALL TRID(TEM(1,1), AA, BB, CC, DD)                              0139000
140*             DO 600 I=1,NR                                                   0140000
141*            P(I,1) = DEN(I,1)*BK*TEM(I,1)                                    0141000
142*            P(I,2) = DEN(I,2)*BK*TEM(I,2)                                    0142000
143*            PTOT(I) = P(I,1) + P(I,2)                                        0143000
144*  600       CONTINUE                                                         0144000
145* C                                                                           0145000
146* C      LIMIT ROUND OFF ERROR AT VACUUM                                      0146000
147*            DO 700 I=1,NR                                                    0147000
148*            FACT(I) = ABS(PTOT(I)/KNEW)                                      0148000
149*            IF(FACT(I) .GT. 1.0E-2) GO TO 700                                0149000
150*            TEM(I,1) = OTEM(I,1)                                             0150000
151*            TEM(I,2) = OTEM(I,2)                                             0151000
152*            P(I,1) = DEN(I,1)*BK*TEM(I,1)                                    0152000
153*            P(I,2) = DEN(I,2)*BK*TEM(I,2)                                    0153000
```

*** MEMBER FLOW

```
          PTOT(I) = P(I,1) + P(I,2)
  700     CONTINUE
C
C       ADVANCE BZ
          DO 101 I=2,NRM
          DELRP = R(I+1) - R(I)
          DELRM = R(I) - R(I-1)
          RATONE = SDT/ DR(I)
          RONE(I) = RATONE/DELRM
          R2(I) = RATONE/DELRP
          RAT2 = RG(I)/R(I)
          RAT3 = RG(I+1)/R(I)
          RAT4 = R(I-1)/R(I)
          RAT5 = R(I+1)/R(I)
          BONE = RONE(I)*BD3(I)*RAT2/2.0
          BTWO = R2(I)*BD3(I+1)*RAT3/2.0
          BTHREE = RONE(I)*BDONE(I)*RAT4
          BFOUR = R2(I)*BDONE(I+1)*RAT5
          AA(I) = BONE
          BB(I) = -(1.0 + BONE + BTWO)
          CC(I) = BTWO
          DD(I) = -BONE*OBZ(I-1) - (1.0 - BONE - BTWO)*OBZ(I) -
     .      BTWO*OBZ(I+1) + BTHREE*OBT(I-1) - (BDONE(I+1)*R2(I) +
     .      BDONE(I)*RONE(I))*OBT(I) + BFOUR*OBT(I+1)
  101     CONTINUE
C
C       B.C. FOR BZ AT R=0
          DELRP = R(2) - R(1)
          R2(1) = SDT/(DR(1)*DELRP)
          RAT3 = RG(2)/R(1)
          BTWO = R2(1)*BD3(2)*RAT3/2.0
          BFOUR = R2(1)*BDONE(2)*R(2)/R(1)
          AA(1) = 0.0
          BB(1) = -(1.0 + BTWO)
          CC(1) = BTWO
          DD(1) = -(1.0 - BTWO)*OBZ(1) - BTWO*OBZ(2) - (R2(1)*BDONE(2))
     .      *OBT(1) + BFOUR*OBT(2)
C
C       B.C. FOR BZ AT WALL
          DELRM = R(NR) - R(NRM)
          RONE(NR) = SDT/(DR(NR)*DELRM)
          RAT2 = RG(NR)/R(NR)
          BONE = RONE(NR)*BD3(NR)*RAT2/2.0
          BTHREE = RONE(NR)*BDONE(NR)*R(NRM)/R(NR)
          AA(NR) = BONE
          BB(NR) = -(1.0 + BONE)
          CC(NR) = 0.0
          DD(NR) = -BONE*OBZ(NRM) - (1.0 - BONE)*OBZ(NR) +
     .      BTHREE*OBT(NRM) - (RONE(NR)*BDONE(NR))*OBT(NR)
          CALL TRID(BZ, AA, BB, CC, DD)
C
```

```
*** MEMBER FLOW

205* C       ADANCE BT FIELD                                                    0205000
206*           DO 102 I=2,NRM                                                   0206000
207*           RAT2 = R(I-1)/RG(I)                                              0207000
208*           RAT3 = R(I)/RG(I)                                                0208000
209*           RAT4 = R(I)/RG(I+1)                                              0209000
210*           RAT5 = R(I+1)/RG(I+1)                                            0210000
211*           BONE = RONE(I)*BD2(I)*RAT2/2.0                                   0211000
212*           BTWO = RONE(I)*BD2(I)*RAT3/2.0                                   0212000
213*           BTHREE = R2(I) * BD2(I+1)*RAT4/2.0                               0213000
214*           BFOUR = R2(I)*BD2(I+1)*RAT5/2.0                                  0214000
215*           AA(I) = BONE                                                     0215000
216*           BB(I) = -(1.0 + BTHREE + BTWO)                                   0216000
217*           CC(I) = BFOUR                                                    0217000
218*           DD(I) = -BONE*OBT(I-1) - (1.0 -BTHREE - BTWO)*OBT(I) - BFOUR*    0218000
219*     .       OBT(I+1) + RONE(I)*BDONE(I)*OBZ(I-1) - (RONE(I)*BDONE(I)       0219000
220*     .       + R2(I)*BDONE(I+1))*OBZ(I) + R2(I)*BDONE(I+1)*OBZ(I+1)         0220000
221*  102      CONTINUE                                                         0221000
222* C                                                                         0222000
223* C       B.C. FOR BT AT R=0                                                0223000
224*           DELRP = R(2) - R(1)                                              0224000
225*           RAT4 = R(1)/RG(2)                                                0225000
226*           RAT5 = R(2)/RG(2)                                                0226000
227*           R2(1) = SDT/(DR(1)*DELRP)                                        0227000
228*           BTHREE = R2(1)*BD2(2)*RAT4/2.0                                   0228000
229*           BFOUR = R2(1)*BD2(2)*RAT5/2.0                                    0229000
230*           DUM = BD2(1)*SDT/DR(1)**2                                        0230000
231*           AA(1) = 0.0                                                      0231000
232*           BB(1) = -(1.0 + BTHREE + DUM)                                    0232000
233*           CC(1) = BFOUR                                                    0233000
234*           DD(1) = -(1.0 - BTHREE-DUM)*OBT(1) - BFOUR*OBT(2)                0234000
235*     .       +BDONE(2)*R2(1)*(OBZ(2) - OBZ(1))                              0235000
236* C                                                                         0236000
237* C       B.C. FOR BT AT WALL                                               0237000
238*           DELRM = R(NP) - R(NRM)                                           0238000
239*           RAT2 = R(NRM)/RG(NR)                                             0239000
240*           RAT3 = R(NR)/RG(NR)                                              0240000
241*           RONE(NR) = SDT/(DR(NR)*DELRM)                                    0241000
242*           BONE = RONE(NR)*BD2(NP)*RAT2/2.0                                 0242000
243*           BTWO = RONE(NR)*BD2(NP)*RAT3/2.0                                 0243000
244*           AA(NR) = BONE                                                    0244000
245*           BB(NR) = -(1.0 + BTWO)                                           0245000
246*           CC(NR) = 0.0                                                     0246000
247*           DD(NR) = -BONE*OBT(NRM) - (1.0 - BTWO)*OBT(NR) -                 0247000
248*     .       BDONE(NRM)*RONE(NR)*(OBZ(NR) - OBZ(NRM))                       0248000
249*           CALL TRID( BT, AA, BB, CC, DD)                                   0249000
250*           BT(1) = BT(2)/2.0                                                0250000
251* C                                                                         0251000
252* C       CONVERT ALL ALL QUANTITIES TO GRID POINTS                         0252000
253*           DO 120 I=2,NR                                                    0253000
254*           BZG(I) = BZ(I-1)*C1(I) + BZ(I)*C2(I)                             0254000
255*           BTG(I) = BT(I-1)*C1(I) + BT(I)*C2(I)                             0255000
```

```
*** MEMBER FLOW

256*            DO 120 J=1,NS                                                   0256000
257*            TEM(I,J) = AMAX1(TEM(I,J),1.1)                                  0257000
258*            TEMG(I,J) = C1(I)*TEM(I-1,J) + C2(I)*TEM(I,J)                   0258000
259*            PG(I,J) = C1(I)*P(I-1,J) + C2(I)*P(I,J)                         0259000
260*  120       CONTINUE                                                        0260000
261*            DO 121 J=1,NS                                                   0261000
262*            TEMG(1,J) = 0.0                                                 0262000
263*            TEMG(NRP,J) = 0.0                                               0263000
264*            PG(1,J) = 0.0                                                   0264000
265*  121       PG(NRP,J) = 0.0                                                 0265000
266*            BZG(NRP) = BZ(NR)                                               0266000
267*            BZG(1) = -SQRT(KNEW*PI8)                                        0267000
268*            BTG(1) = 0.0                                                    0268000
269*            BTG(NRP) = BT(NR)*R(NR)/RG(NRP)                                 0269000
270*            DO 123 I=1,NRP                                                  0270000
271*  123       PTOTG(I) = PG(I,1) + PG(I,2)                                    0271000
272* C       THIS ELIMINATES ITERATION                                          0272000
273*          GO TO 2                                                         0273000
274*            IF(NSTP .EQ. 1) GO TO 5                                         0274000
275*            DO 401 I=1,NR                                                   0275000
276*            IF(FACT(I) .LT. 1.0E-2) GO TO 401                               0276000
277*            DTEME(I) = ABS((TTEME(I) - TEM(I,1))/TEM(I,1))                  0277000
278*            DTEMI(I) = ABS((TTEMI(I) - TEM(I,2))/TEM(I,2))                  0278000
279*            DBZ(I) = ABS((TBZ(I) - BZ(I))/BZ(I))                            0279000
280*            DBT(I) = ABS((TBT(I) - BT(I))/BT(I))                            0280000
281*  401       CONTINUE                                                        0281000
282*            K = MAXVAL(DTEME) + 1                                           0282000
283*            L = MAXVAL(DTEMI) + 1                                           0283000
284*            M = MAXVAL(DBZ) + 1                                             0284000
285*            N = MAXVAL(DBT) + 1                                             0285000
286*            TEST = AMAX1(DTEME(K), DTEMI(L), DBZ(M), DBT(N))                0286000
287*            IF(TEST .LE. 5.0E-2) GO TO 2                                    0287000
288*  5         CONTINUE                                                        0288000
289*            DO 402 I=1,NR                                                   0289000
290*            TTEMI(I) = TEM(I,2)                                             0290000
291*            TTEME(I) = TEM(I,1)                                             0291000
292*            TBZ(I) = BZ(I)                                                  0292000
293*            TBT(I) = BT(I)                                                  0293000
294*  402       CONTINUE                                                        0294000
295*            IF(NSTP .LT. 10) GO TO 1                                        0295000
296*            WRITE(6,405)(TEM(I,1), TEM(I,2), BZ(I), BT(I), DTEME(I),        0296000
297*     .        DTEMI(I), DBZ(I), DBT(I), I=1,NR)                             0297000
298*  405       FORMAT(1X, 8(E12.4,2X))                                         0298000
299*            STOP                                                            0299000
300*  2         CONTINUE                                                        0300000
301*            CALL FLUX                                                       0301000
302*            CALL LAGRAG                                                     0302000
303*          RETURN                                                            0303000
304*          END                                                               0304000
```

```
*** MEMBER FLUX

      SUBROUTINE FLUX
C
C  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C     FLUX COMPUTES THE ADIABATIC INVARIANT QUANTITIES
C     MASS    = TOTAL MASS OF SPECIE J IN EACH CELL
C       BZFLX = BZ FLUX              ,  S  =  ADIABATIC GAMMA = 3/2
C  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C
      IMPLICIT REAL*4 (A - Z)
      VDIM
      COMMON/CON1/ PI, PI2, PI4, PI6, PI8, BK, C
      COMMON/GRID/ RG(NPP), R(NR), C1(NRP), C2(NRP), DR(NRP)
      COMMON/SPECIE/ AM(NS), Q(NS)
      COMMON/MTM/ BZ(NR), DEN(NR,NS), TEM(NR,NS), P(NR,NS), PTOT(NR)
     .   , BT(NR)
      COMMON/FLUXO/ MASS(NRP,NS), S(NRP,NS), BZFLX(NRP),BTFLX(NRP)
      COMMON/CON2/ GAM, KNEW
      INTEGER I,J
C
      REAL*4 RHO(NR,NS)
C
C     CALCULATE THE FLUXES
      DO 100 I=1,NR
      RSDIF = RG(I+1)**2 - RG(I)**2
      RSDIFP = RSDIF*PI
      RHO(I,1) = DEN(I,1)*AM(1)
      RHO(I,2) = DEN(I,1)*AM(2)
      MASS(I,1) = RHO(I,1)*RSDIFP
      MASS(I,2) = RHO(I,2)*RSDIFP
      DENK = DEN(I,1)*BK
      P(I,1) = DENK*TEM(I,1)
      P(I,2) = DENK*TEM(I,2)
      S(I,1) = P(I,1)/RHO(I,1)**GAM
      S(I,2) = P(I,2)/RHO(I,2)**GAM
      BZFLX(I) = RSDIFP*BZ(I)
      BTFLX(I) = BT(I)*DR(I)
100   CONTINUE
      RETURN
      END
```

```
*** MEMBER INCON

      1*            SUBROUTINE INCON (DRG,ETA,T0,BZ0,DEN0,RWALL,BT0,          0001000
      2*           .    RJP, DRJZ, DRJT )                                     0002000
      3* C                                                                    0003000
      4* C     INCON COMPUTES INITIAL CONDITIONS FOR LINER SETUP               0004000
      5* C                                                                    0005000
      6* C                                                                    0006000
      7*            IMPLICIT REAL*4 (A -Z)                                    0007000
      8*            PARAMETER NRP=29, NS=2, NR=NRP-1                          0008000
      9*            COMMON/BC/ NFRAC, TFRAC, TLBASE, TRBASE                   0009000
     10*            COMMON/ONE/ F0, BFLUX                                     0010000
     11*            COMMON/TWO/ TOTE, VOL                                     0011000
     12*            COMMON/FOUR/ CDI                                          0012000
     13*            COMMON/BASE/ TBASE, NBASE                                 0013000
     14*            COMMON/CON1/ PI, PI2, PI4, PI6, PI8, BK, C                0014000
     15*            COMMON/GRID/ RG(NRP), R(NR), C1(NRP), C2(NRP), DR(NRP)    0015000
     16*            COMMON/SPECIE/ AM(NS), Q(NS)                              0016000
     17*            COMMON/MTM/ BZ(NR), DEN(NR,NS), TEM(NR,NS), P(NR,NS), PTOT(NR) 0017000
     18*           .          , BT(NR)                                        0018000
     19*            COMMON/MTG/ BZG(NRP), DENG(NRP,NS), TEMG(NRP,NS), PG(NRP,NS), 0019000
     20*           .          PTOTG(NRP), BTG(NRP)                            0020000
     21*            COMMON/CON2/ GAM, KNEW                                    0021000
     22*            COMMON/OLD/ LOCRET(NRP), ROLD(NP), DROLD(NR), DENOLD(NR), 0022000
     23*           .    TEMOLD(NR,NS), PTOTO(NR), BZOLD(NR), COLD, WOLD,WALLE,BTOLD 0023000
     24*           .    , DIAMO                                               0024000
     25*            COMMON/OUT2/ BASE                                         0025000
     26*            COMMON/MAXV/ BZMAX, BTMAX,BPMAX, PTMAX, TEMAX, TIMAX      0026000
     27* C                                                                    0027000
     28*            REAL*4 RHO(NR,NS), BPRE(NR)                               0028000
     29*            INTEGER I, J, K, L, II                                    0029000
     30*            INTEGER JJ, KK, LL, MM, QQ                                0030000
     31*  1000      FORMAT(//4X,'SPECIE ',I2,9X,'Q= ',1PG12.4,9X,'M= ',1PG12.4) 0031000
     32* 2000       FORMAT(//4X, 'INITIAL BETA =',1PG12.4,                    0032000
     33*           .    10X,'TOTAL ENERGY=',1PG12.4,10X,'TOTAL PRESSURE CONST=', 0033000
     34*           .    1PG12.4)                                             0034000
     35* C                                                                    0035000
     36* C                                                                    0036000
     37*            BASE = -1.0                                               0037000
     38*            DO 100 I=1,NRP                                            0038000
     39*            RG(I) = DRG*FLOAT(I-1)                                    0039000
     40*  100        CONTINUE                                                 0040000
     41*            DO 200 I=1,NR                                             0041000
     42*  200       R(I) = (RG(I) + RG(I+1))*0.5                              0042000
     43* C                                                                    0043000
     44* C      STORE GRID INFORMATION FOR LATER ADIABATIC CALCULATION         0044000
     45* C                                                                    0045000
     46*            DO 210 I=1,NR                                             0046000
     47*            ROLD(I) = R(I)                                            0047000
     48*            DROLD(I) = RG(I+1) - RG(I)                                0048000
     49*            DR(I) = DROLD(I)                                          0049000
     50*  210       CONTINUE                                                  0050000
     51*            DO 220 I=2,NR                                             0051000
```

```
*** MEMBER INCON

      52*           C1(I) = DR(I)/(DR(I) + DR(I-1))                          0052000
      53*           C2(I) = 1.0 -C1(I)                                       0053000
      54*  220      CONTINUE                                                 0054000
      55* C                                                                  0055000
      56* C      INITIAL PROFILES AT MID POINTS                              0056000
      57*           DO 303 I=2,NRP                                           0057000
      58*           BS = 0.0                                                 0058000
      59*           II = I -1                                                0059000
      60*           DO 304 K=1,II                                            0060000
      61*  304       BS= BS + EXP(-(R(K)-RJP)**2/DRJZ)*R(K)*DR(K)            0061000
      62*           BTG(I) = BT0*BS/RG(I)                                    0062000
      63*  303      CONTINUE                                                 0063000
      64*           BTG(1) = 0.0                                             0064000
      65*           DO 300 I=1,NR                                            0065000
      66*  300      BT(I) = (BTG(I) + BTG(I+1))/2.0                          0066000
      67*           BTSUM = 0.0                                              0067000
      68*           DO 302 I=1,NR                                            0068000
      69*  302      BTSUM = BTSUM + BT(I)**2/(PI4*R(I))*DR(I)                0069000
      70*           KNEW = (BZ0**2 + BTG(NRP)**2)/PI8 + BTSUM                0070000
      71*           BZ0L = SQRT(KNEW*PI8)                                    0071000
      72*           A = (BZ0 +BZ0L)/(SQRT(PI*DRJT))                          0072000
      73*           DO 308 I=2,NRP                                           0073000
      74*           BS = 0.0                                                 0074000
      75*           II = I - 1                                               0075000
      76*           DO 309 K=1,II                                            0076000
      77*  309       BS=BS + EXP(-(R(K)-RJP)**2/DRJY)*DR(K)                  0077000
      78*           BS = BS*A                                                0078000
      79*           BZG(I) = -BZ0L + BS                                      0079000
      80*  308      CONTINUE                                                 0080000
      81*           BZG(1) = -BZ0L                                           0081000
      82*           BZG(NRP) = BZ0                                           0082000
      83*           DO 312 I=1,NR                                            0083000
      84*  312      BZ(I) = (BZG(I) + BZG(I+1))/2.0                          0084000
      85*           DO 305 I=1,NR                                            0085000
      86*           II = I -1                                                0086000
      87*           BST = 0.0                                                0087000
      88*           IF(I .EQ. 1) GO TO 306                                   0088000
      89*           DO 307 K=1,II                                            0089000
      90*  307      BST = BST + BT(K)**2*DR(K)/(R(K)*PI4)                    0090000
      91*  306      BTH = BT(I)**2*DR(I)/(R(I)*PI8)                          0091000
      92*           BPRE(I) = (BZ(I)**2 + BT(I)**2)/PI8 + BST + BTH          0092000
      93*  305      CONTINUE                                                 0093000
      94*           DO 311 I=1,NR                                            0094000
      95*           PTOT(I) = ABS(KNEW - BPRE(I))                            0095000
      96*           TEM(I,1) = T0*SIN(PI*R(I)/4.0)                           0096000
      97*           TEM(I,1) = AMAX1(TEM(I,1),0.5)                           0097000
      98*           TEM(I,2) = TEM(I,1)*ETA                                  0098000
      99*           DEN(I,1) = PTOT(I)/(BK*(TEM(I,1) + TEM(I,2)))            0099000
     100*           DEN(I,2) = DEN(I,1)                                      0100000
     101*           DENK = DEN(I,1)*BK                                       0101000
     102*           P(I,1) = DENK*TEM(I,1)                                   0102000
```

```
*** MEMBER INCON

103*            P(I,2) = DENK*TEM(I,2)                                          0103000
104*  311       CONTINUE                                                        0104000
105* C                                                                         0105000
106* C       INITIAL PROFILES AT GRID POINTS                                   0106000
107*            DO 500 I=2,NR                                                  0107000
108*            DENG(I,1) = DEN(I-1,1)*C1(I) + DEN(I,1)*C2(I)                  0108000
109*            DENG(I,2) = DENG(I,1)                                          0109000
110*            TEMG(I,1) = TEM(I-1,1)*C1(I) + TEM(I,1)*C2(I)                  0110000
111*            TEMG(I,2) = TEMG(I,1)*ETA                                      0111000
112*            PG(I,1) = DENG(I,1)*BK*TEMG(I,1)                               0112000
113*            PG(I,2) = DENG(I,2)*BK*TEMG(I,2)                               0113000
114*            PTOTG(I) = PG(I,1) + PG(I,2)                                   0114000
115*            LOGBET(I) = PTOTG(I)/BPRE(I)                                   0115000
116*  500       CONTINUE                                                       0116000
117*            PTOTG(NRP) = 0.0                                               0117000
118*            PTOTG(1) = 0.0                                                 0118000
119*            DO 501 J=1,NS                                                  0119000
120*            PG(1,J) = 0.0                                                  0120000
121*            PG(NRP,J) = 0.0                                                0121000
122*            TEMG(1,J) = 0.0                                                0122000
123*            TEMG(NRP,J) = TRBASE                                           0123000
124*            DENG(1,J) = DEN(1,J)                                           0124000
125*            DENG(NRP,J) = DEN(NR,J)                                        0125000
126*  501       CONTINUE                                                       0126000
127*            LOGRET(1) = PTOTG(1)/BPRE(1)                                   0127000
128*            LOGRET(NRP) = PTOTG(NRP)/BPRE(NR)                              0128000
129* C                                                                         0129000
130* C       CALCULATE TOTAL ENERGY TOTE                                       0130000
131*            TOTE = 0.0                                                     0131000
132*            VOL = 0.0                                                      0132000
133*            DO 700 I=1,NR                                                  0133000
134*            TOTE = TOTE + PI2*R(I)*DR(I)*(1.5*DEN(I,1)*BK*(TEM(I,1) +      0134000
135*     .        TEM(I,2)) + BZ(I)**2/PI8 + BT(I)**2/PI8)                     0135000
136*            VOL = VOL + PI2*R(I)*DR(I)                                     0136000
137*  700       CONTINUE                                                       0137000
138* C                                                                         0138000
139* C       STORE PROFILES INFORMATION FOR OUTPUT                             0139000
140*            F0 = 0.0                                                       0140000
141*            DO 600 I=1,NR                                                  0141000
142*            DENOLD(I) = DEN(I,1)                                           0142000
143*            TEMOLD(I,1) = TEM(I,1)                                         0143000
144*            TEMOLD(I,2) = TEM(I,2)                                         0144000
145*            PTOTO(I) = PTOT(I)                                             0145000
146*            BZOLD(I) = BZ(I)                                               0146000
147*            F0 = F0 + PI2*R(I)*DR(I)*BZ(I)                                 0147000
148*  600       CONTINUE                                                       0148000
149*            JJ = MAXVAL(BPRE) + 1                                          0149000
150*            KK= MAXVAL(PTOT) + 1                                           0150000
151*            LL = MAXVAL(TEM(*,1)) + 1                                      0151000
152*            MM = MAXVAL(TEM(*,2)) + 1                                      0152000
153*            QQ = MINVAL(BT) + 1                                            0153000
```

*** MEMBER INCON

```
154*            BZMAX = BZ(1)                                          0154000
155*            BTMAX = BT(QQ)                                         0155000
156*            PTMAX = PTOT(KK)                                       0156000
157*            TEMAX = TEM(LL,1)                                      0157000
158*            TIMAX = TEM(MM,2)                                      0158000
159*            BPMAX = BPRE(JJ)                                       0159000
160*            BTOLD = BT(23)                                         0160000
161*            DIAMO = 0.0                                            0161000
162*            DO 900 I=1,23                                          0162000
163*            DIAMO = DIAMO + PI2*R(I)*DR(I)*BZ(I)                   0163000
164*  900       CONTINUE                                               0164000
165*            BFLUX = FO                                             0165000
166* C                                                                 0166000
167* C       INITIAL AKNEW(PLASMA AND MAGNETIC PRESSURE)               0167000
168*            COLD = KNEW                                            0168000
169*            WOLD = RWALL                                           0169000
170*            WALLE = 0.0                                            0170000
171*            DO 550 I=1,NS                                          0171000
172*            WRITE(6,1000) I, Q(I), AM(I)                           0172000
173*  550       CONTINUE                                               0173000
174*            WRITE(6,2000) LOCBET(1), TOTE, KNEW                    0174000
175*            RETURN                                                 0175000
176*            END                                                    0176000
```

```
*** MEMBER LAGRAG

      SUBROUTINE LAGRAG
C
C     LAGRAG COMPUTES THE EQUILIBRIUM POSITIONS OF GRIDS USING
C     LAGRANGIAN METHODS
C     DELR = DISPLACEMENTS OF GRIDS
C     RATIO = DISPLACEMENTS/GRIDS SPACING
C
         IMPLICIT REAL*4 (A-Z)
*        VDIM
         COMMON/RWAL/ RWALL
         COMMON/CON2/ GAM, KNEW
         COMMON/GRID/ RG(NRP), R(NR), C1(NRP), C2(NRP), DR(NRP)
         REAL*4 DELR(NRP), DELR1(NRP), RATIO(NR)
         INTEGER I,J
C
 5       KOLD = KNEW
         TESTK1 = KOLD*1.1
         CALL DDELR (TESTK1, KOLD, DELR)
         DO 100 I=1, NRP
100      DELR1(I) = DELR(I)
         TESTK2 = KOLD*0.9
         CALL DDELR(TESTK2, KOLD, DELR)
         C3 = (RWALL - RG(NRP) - DELR(NRP))/(DELR1(NRP) - DELR(NRP))
         C4 = 1.0- C3
         KNEW = C3*TESTK1 + C4*TESTK2
         DO 200 I=2,NRP
200      DELR(I) = C3*DELR1(I) + C4*DELR(I)
         RG(2) = RG(2) + DELR(2)
         IF(RG(2) .LT. RG(1)) RG(2) = RG(NRP)/FLOAT(NR)
         DO 300 I=3,NRP
         RG(I) = RG(I) + DELR(I)
         RMID = ABS(RG(I) - RG(I-1))/2.0 + RG(I-1)
         RG(I) = AMAX1(RMID, RG(I))
300      CONTINUE
         DO 310 I=1,NR
         DR(I) = RG(I+1) - RG(I)
         RATIO(I) = ABS(DELR(I)/DR(I))
310      CONTINUE
         J = MAXVAL(RATIO) + 1
         IF (RATIO(J) .GT. 5.0) STOP
         IF (RATIO(J) .GT. 1.0E-4) GO TO 5
400       CONTINUE
         DO 700 I=1,NR
         R(I) = (RG(I) + RG(I+1))/2.0
700      CONTINUE
         RETURN
         END
```

*** MEMBER LINER

```
      PROGRAM LINER                                                          0001000
      IMPLICIT REAL*4 (A - Z)                                                0002000
      PARAMETER NRP=29, NS=2, NR=NRP-1                                       0003000
      COMMON/BC/ NFRAC, TFRAC, TLBASE, TRBASE                                0004000
      COMMON/BASE/ TBASE, NBASE                                              0005000
      COMMON/DUMP/ LDUMP                                                     0006000
      COMMON/CON2/ GAM, KNEW                                                 0007000
      COMMON/SPECIE/ AM(NS), Q(NS)                                           0008000
      COMMON/CON1/ PI, PI2, PI4, PI6, PI8, BK, C                             0009000
      COMMON/SWITCH/ LPRINT, LDSP, LWALL, LPLOT, LBOOK                       0010000
      COMMON/RWAL/ RWALL                                                     0011000
      COMMON/HANDLE/ LTURB                                                   0012000
      INTEGER I, J, NSTEP, NPRINT, NSTOP, NPLOT                              0013000
      LOGICAL LPRINT, LDSP, LWALL, LPLOT, LBOOK, LTURB, LDUMP                0014000
C                                                                            0015000
C  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *    0016000
C     UNITS OF THIS CODE ARE                                                 0017000
C        X = CM                                                              0018000
C        DENSITY = NUMBER/CM**3                                              0019000
C        VELOCITY = CM/SEC                                                   0020000
C        TEMP = EV                                                           0021000
C        B = GAUSS                                                           0022000
C        E = STATVOLT/CM                                                     0023000
C  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *    0024000
C                                                                            0025000
C                                                                            0026000
      NAMELIST/INPUT/ GAM, DRG, ETA, R0, ALFA, DEN0, BZ0, TFACT,             0027000
     .   B1,RRIGHT, RMIN, TMID, T0, DT, RWALL, NPRINT, NSTOP                 0028000
     .   , BT0, TLBASE, TRBASE, NFRAC, TFRAC, NPLOT                          0029000
     .,RJP,DRJZ,DRJT                                                         0030000
      NAMELIST/KNOB/ LPRINT, LDSP, LWALL, LPLOT, LBOOK, LTURB                0031000
     .    , LDUMP                                                           0032000
      DATA LPRINT, LPLOT/ 2*.TRUE./, LDSP, LWALL/2*.FALSE./,                 0033000
     .   LBOOK/.FALSE./, LTURB/.FALSE./, LDUMPY.FALSE./                     0034000
      DATA PI/3.14159/, PI2/6.28319/, PI4/12.5664/, PI6/18.8496/,            0035000
     .   PI8/25.1327/, C/3.0E10/, BK/1.6022E-12/, AM(1)/9.10904E-28/,        0036000
     .   AM(2)/3.345E-24/, Q(1)/-4.80296E-10/, Q(2)/4.80296E-10/             0037000
      DATA GAM/1.5/, DRG/1.0/, R0/30.0/, ALFA/25.0/, DEN0/1.0E14/,           0038000
     .   BZ0/1.0E3/, TFACT/5.0/, RRIGHT/50.0/, RMIN/10.0/,                   0039000
     .   TMID/1.E-3/, NPRINT/1/, NSTOP/1/, ETA/10.0/, B1/100.0/,            0040000
     .   RWALL/30.0/, DT/5.0E-5/, T0/100.0/, FRAC/1.0E3/,                   0041000
     .   TBASE/50.0/, NBASE/1.0E12/, NFRAC/1.0E-4/, TFRAC/0.8/,             0042000
     .   TLBASE/20.0/, TRBASE/2.0/, NPLOT/10/                               0043000
C                                                                            0044000
      CALL RSSTOP                                                            0045000
      READ (5,INPUT)                                                         0046000
      WRITE(6,INPUT)                                                         0047000
      READ (5,KNOB)                                                          0048000
      WRITE (6,KNOB)                                                         0049000
      IF( .NOT. LBOOK) GO TO 100                                             0050000
       CALL INCON (DRG,ETA,T0,BZ0,DEN0,RWALL,BT0,                            0051000
```

```
*** MEMBER LINER

      52*     .    RJP,DRJZ,DRJT)                                                    0052000
      53*          GO TO 200                                                         0053000
      54*  100     CONTINUE                                                          0054000
      55*          CALL TINCON(DRG, ETA, R0, ALFA, DEN0, BZ0, B1,T0)                 0055000
      56*  200      CONTINUE                                                         0056000
      57*          CALL FLUX                                                         0057000
      58*          CALL LAGRAG                                                       0058000
      59*          CALL OUTPUT(0.0, 0.0, 0, NPLOT)                                   0059000
      60*          CALL STEP(FRAC, TFACT, RRIGHT, RMIN, TMID, DT,                    0060000
      61*     .      NPRINT, NSTOP, NPLOT)                                          0061000
      62*          STOP                                                              0062000
      63*          END                                                               0063000
```

*** MEMBER OUTPUT

```
      1*       SUBROUTINE OUTPUT(CLOCK, DT, NSTEP, NPLOT)                                0001000
      2* C                                                                              0002000
      3* C     OUTPUT DIRECTS ALMOST ALL OUTPUT WHICH INCLUDES PLOTTING                  0003000
      4* C                                                                              0004000
      5*         IMPLICIT REAL*4 (A -Z)                                                 0005000
      6*         PARAMETER NRP=29, NS=2, NR=NRP-1                                       0006000
      7*         COMMON/BC/ NFRAC, TFRAC, TLBASE, TRBASE                                0007000
      8*         COMMON/ONE/ E0, BFLUX                                                  0008000
      9*         COMMON/TWO/TOTE, VOL                                                   0009000
     10*         COMMON/RWAL/ RWALL                                                     0010000
     11*         COMMON/OUT1/ TAUG(NRP,NS), HT(NRP,NS), QERU(NRP), QERT(NRP),           0011000
     12*      .    SQEG(NRP), SQIG(NRP), DELVG(NRP), EG(NRP), WCG(NRP,NS)               0012000
     13*      .     , VCOLG(NRP)                                                       0013000
     14*         COMMON/CON1/ PI, PI2, PI4, PI6, PIA, BK, C                             0014000
     15*         COMMON/CON2/GAM, KNEW                                                  0015000
     16*         COMMON/SPECIE/ AM(NS), Q(NS)                                           0016000
     17*         COMMON/GRID/ RG(NRP), R(NR), C1(NRP), C2(NRP), DR(NRP)                 0017000
     18*         COMMON/FLUXG/ MASS(NRP,NS), S(NRP,NS), BZFLX(NRP),BTFLX(NRP)           0018000
     19*         COMMON/MTM/ BZ(NR), DEN(NR,NS), TEM(NR,NS), P(NR,NS), PTOT(NR)         0019000
     20*      .    , BT(NR)                                                           0020000
     21*         COMMON/MTG/ BZG(NRP), DENG(NRP,NS), TEMG(NRP,NS), PG(NRP,NS),          0021000
     22*      .    PTOTG(NRP), BTG(NRP)                                               0022000
     23*         COMMON/SWITCH/ LPRINT,LDSP, LWALL, LPLOT, LBOOK                        0023000
     24*         COMMON/OLD/LOCBET(NRP), BOLD(NR), DROLD(NR), DENOLD(NR),               0024000
     25*      .    TEMOLD(NR,NS), PTOTO(NR), BZOLD(NR), COLD, WOLD,WALLE,BTOLD        0025000
     26*      .    , DIAMO                                                            0026000
     27*         COMMON/OUT2/ BASE                                                      0027000
     28*         COMMON/HYBRID/ VTG(NRP,NS), WLHG(NRP), VCRIT(NRP), VD(NRP),            0028000
     29*      .    ANOV(NRP), ANOHT(NRP), WPG(NRP,NS)                                 0029000
     30*         COMMON/HANDLE/ LTURB                                                   0030000
     31*         COMMON/MAXV/BZMAX, BTMAX, BPMAX, PTMAX, TEMAX, TIMAX                   0031000
     32*         REAL*4 BPRE(NR), PLOT(NR,7), CONST(NRP), VEB(NRP), DENT(NR),           0032000
     33*      .    TEMT(NR,NS), PTOTT(NR), BZT(NR), W(NR), DV(NR), NUM(NR)            0033000
     34*      .    , BUFF(1000), CPLOTF(NRP), CPLOTB(NRP)                             0034000
     35*      .    , CPLOTE(NRP), CPLOTI(NRP), CPLOTN(NRP), CPLOTH(NRP)               0035000
     36*      .    , JT(NR), JZ(NR), DELV(NR), DELVZ(NR), RESIST(NR)                  0036000
     37*         INTEGER JTE, JTI                                                       0037000
     38*         INTEGER K, L, M, I, J, MM, NSTEP, NPLOT, II                            0038000
     39*         INTEGER KK, LL, JJ                                                     0039000
     40*         LOGICAL LPRINT, LDSP, LWALL, LPLOT, LBOOK,LTURB, LCALCM                0040000
     41* C                                                                              0041000
     42* 1000    FORMAT(1H1,'TIME = ',1PG12.2,2X,'TIME-STEP = ',1PG12.2,2X,             0042000
     43*      .    'STEP = ', I10/)                                                   0043000
     44* 2000    FORMAT(1X,'GRID',6X,'R',10X,'B',12X,'N',10X,'TE',10X,'TI',             0044000
     45*      .    10X,'PTOTAL',8X,'BP', 8X, 'CONSTANT',8X,'BT')                      0045000
     46* 2010    FORMAT(1X, I3, 9E12.4)                                                 0046000
     47* 2020    FORMAT(//1X,'GRID',5X,'R',6X,'TAU-E',6X,'Q-F',6X,'QERU',6X,            0047000
     48*      .    'QERT',3X,'E-HFLX',6X,'Q-I',6X,'I-HFLX',6X,'WCE',6X,               0048000
     49*      .    'VCOL', 4X, 'VEB'/)                                                0049000
     50* 2030    FORMAT(1X, I3, 11E10.2)                                                0050000
     51* 2060    FORMAT(1X,'GRID', 6X, 'R', 10X, 'JZ', 11X, 'JT', 10X,                  0051000
```

```
*** MEMBER OUTPUT

     52*          .     'DELV', 8X, 'DELVZ', 7X, 'RESIST')                    0052000
     53*  2070      FORMAT(1X,I3,6E12.4)                                      0053000
     54*  C                                                                   0054000
     55*  C                                                                   0055000
     56*  C       CALCULATE B, N, T FROM FLUXES                               0056000
     57*  C                                                                   0057000
     58*  C        CALCULATE PROFILES AT MID POINTS                           0058000
     59*             DO 305 I=1,NR                                           0059000
     60*             RSDIF = RG(I+1)**2 - RG(I)**2                           0060000
     61*             RSDIFP = RSDIF*PI                                       0061000
     62*             BZ(I) = BZFLX(I)/(PI*RSDIF)                             0062000
     63*             BT(I) = BTFLX(I)/DR(I)                                  0063000
     64*             DEN(I,1) = MASS(I,1)/(AM(1)*RSDIFP)                     0064000
     65*             DEN(I,2) = DEN(I,1)                                     0065000
     66*             P(I,1) = S(I,1)*(DEN(I,1)*AM(1))**GAM                   0066000
     67*             P(I,2) = S(I,2)*(DEN(I,2)*AM(2))**GAM                   0067000
     68*             TEM(I,1) = P(I,1)/(BK*DEN(I,1))                         0068000
     69*             TEM(I,2) = P(I,2)/(BK*DEN(I,2))                         0069000
     70*             TEM(I,1) = AMAX1(TEM(I,1),0.1)                          0070000
     71*             TEM(I,2) = AMAX1(TEM(I,2),0.1)                          0071000
     72*             PTOT(I) = BK*DEN(I,1)*(TEM(I,1) + TEM(I,2))             0072000
     73*             PTOT(I) = P(I,1) + P(I,2)                               0073000
     74*             DR(I) = RG(I+1) -RG(I)                                  0074000
     75*             NUM(I) = DEN(I,1)                                       0075000
     76*  305        CONTINUE                                                0076000
     77*  110        CONTINUE                                                0077000
     78*             DO 400 I=2,NR                                           0078000
     79*             C1(I) = DR(I)/(DR(I) + DR(I-1))                         0079000
     80*             C2(I) = 1.0 - C1(I)                                     0080000
     81*  400        CONTINUE                                                0081000
     82*             DO 200 I=1,NR                                           0082000
     83*             II = I - 1                                              0083000
     84*             BST = 0.0                                               0084000
     85*             IF(I .EQ. 1) GO TO 201                                  0085000
     86*             DO 202 K=1,II                                           0086000
     87*  202        BST = BST + BT(K)**2*DR(K)/(R(K)*PI4)                   0087000
     88*  201        BTH = BT(I)**2*DR(I)/(R(I)*PI8)                         0088000
     89*             BPRE(I) = (BZ(I)**2 + BT(I)**2)/PI8 + BST + BTH         0089000
     90*  200        CONTINUE                                                0090000
     91*  C                                                                   0091000
     92*  C       CONVERT ALL ALL QUANTITIES TO GRID POINTS                   0092000
     93*             DO 120 I=2,NR                                           0093000
     94*             BZG(I) = BZ(I-1)*C1(I) + BZ(I)*C2(I)                    0094000
     95*             BTG(I) = BT(I-1)*C1(I) + BT(I)*C2(I)                    0095000
     96*             DO 120 J=1,NS                                           0096000
     97*             DENG(I,J) = C1(I)*DEN(I-1,J) + C2(I)*DEN(I,J)           0097000
     98*             TEMG(I,J) = C1(I)*TEM(I-1,J) + C2(I)*TEM(I,J)           0098000
     99*             PG(I,J) = C1(I)*P(I-1,J) + C2(I)*P(I,J)                 0099000
    100*  120        CONTINUE                                                0100000
    101*             DO 121 J=1,NS                                           0101000
    102*             DENG(1,J) = DEN(1,J)                                    0102000
```

```
*** MEMBER OUTPUT

103*            DENG(NRP,J) = DEN(NR,J)                                          0103000
104*            TEMG(1,J) = 0.0                                                  0104000
105*            TEMG(NRP,J) = 0.0                                                0105000
106*            PG(1,J) = 0.0                                                    0106000
107*  121       PG(NRP,J) = 0.0                                                  0107000
108*            BZG(NRP) = BZ(NR)                                                0108000
109*            BZG(1) = -SQRT(PI8*KNEW)                                         0109000
110*            BTG(1) = 0.0                                                     0110000
111*            BTG(NRP) = BT(NP)*R(NR)/RG(NRP)                                  0111000
112*            DO 123 I=1,NRP                                                   0112000
113*  123       PTOTG(I) = PG(I,1) + PG(I,2)                                     0113000
114* C                                                                           0114000
115* C       CALCULATE QUANTITIES TO BE OUTPUT                                   0115000
116*            ITOTZ = 0.0                                                      0116000
117*            ITOTT = 0.0                                                      0117000
118*            CP = C**2/PI4                                                    0118000
119*            MOE2 = AM(1)/Q(1)**2                                             0119000
120*             LAM= 15.0                                                       0120000
121*            DO 500 I=1,NR                                                    0121000
122*            RESIST(I) = MOE2/(3.5E5*TEM(I,1)**1.5/LAM)                       0122000
123*            JT(I) = -C*(BZG(I+1) - BZG(I))/(DR(I)*PI4)                       0123000
124*            JZ(I) = C/PI4*(RG(I+1)*BTG(I+1) - RG(I)*BTG(I))/(R(I)*DR(I))     0124000
125*            DELV(I) = JT(I)/(DEN(I,1)*Q(1))                                  0125000
126*            DELVZ(I) = JZ(I)/(DEN(I,1)*Q(1))                                 0126000
127*  500       CONTINUE                                                         0127000
128*            B2SUM = 0.0                                                      0128000
129*            DO 520 I=1,NR                                                    0129000
130*            CONST(I) = BPRE(I) + PTOT(I)                                     0130000
131*            B2SUM = B2SUM + BPRE(I)                                          0131000
132*  520       CONTINUE                                                         0132000
133* C                                                                           0133000
134* C       PLOT B, B PRESSURE, PLASMA PRESSURE AND DENSITY                     0134000
135*            IF (.NOT. LPRINT) GO TO 600                                      0135000
136*            IF(.NOT. LPLOT) GO TO 100                                        0136000
137*            L = MAXVAL(NUM) + 1                                              0137000
138*            M1=MINVAL(BT)+1                                                  0138000
139*           BTMAX=BT(M1)                                                      0139000
140*           M1=MAXVAL(BZ) +1                                                  0140000
141*           BZMAX=BZ(M1)                                                      0141000
142*           M1=MAXVAL(BPRE)+1                                                 0142000
143*           BPMAX=BPRE(M1)                                                    0143000
144*           M1=MAXVAL(PTOT)+1                                                 0144000
145*           PTMAX=PTOT(M1)                                                    0145000
146*            DO 300 I=1,NR                                                    0146000
147*            PLOT(I,1) = BT(I)/BTMAX                                          0147000
148*            PLOT(I,2) = BZ(I)/ABS(BZMAX)                                     0148000
149*            PLOT(I,3) = DEN(I,1)/DEN(L,1)                                    0149000
150*            PLOT(I,4) = BPRE(I)/BPMAX                                        0150000
151*            PLOT(I,5) = PTOT(I)/PTMAX                                        0151000
152*            PLOT(I,6) = TEM(I,1)/TEMAX                                       0152000
153*            PLOT(I,7) = TEM(I,2)/TIMAX                                       0153000
```

*** MEMBER OUTPUT

```
          CPLOTN(I) = PLOT(I,1)
          CPLOTE(I) = PLOT(I,6)
          CPLOTI(I) = PLOT(I,7)
          CPLOTH(I) = PLOT(I,2)
          CPLOTP(I) = PLOT(I,5)
          CPLOTB(I) = PLOT(I,4)
  300     CONTINUE
          CALL BSCALE (0.0, R(NR), 0.0, 1.0)
          CALL BPLOT ( R, PLOT(1,3), NR, 'N    $')
          CALL BPLOT (R, PLOT(1,4), NR, 'H    $')
          CALL BPLOT (R, PLOT(1,5), NR, 'PP  EQUILIBRIUM PROFILES$')
          CALL BSCALE(0.0, R(NR), -1.0, 1.0)
          CALL BPLOT(R, PLOT(1,2), NR, 'B    $')
          CALL BPLOT(R, PLOT(1,1), NR, 'H    $')
          CALL BPLOT(R, PLOT(1,6), NR, 'E    $')
          CALL BPLOT(R, PLOT(1,7), NR, 'II  EQUILIBRIUM PROFILES$')
          LCALCM = MOD(NSTEP,NPLOT) .EQ. 0
          IF(.NOT. LCALCM) GO TO 100
          CALL PLOTS(BUFF, 1000, 0.0)
          CALL LABELS (0.0, R(NR), 0.0, 1.0,
     .      '               LINER PROFILES                 ',
     .      '                  RADIUS(CM)                  ',
     .      ' PLASMA PRESSURE AND MAGNETIC PRESSURE       ')
          CALL BOOKPL (R, CPLOTP, NR, 1)
          CALL BOOKPL ( R, CPLOTB, NR, -1)
          CALL LABELS (0.0, R(NR), -1.0, 1.0,
     .      '               LINER PROFILES                 ',
     .      '                  RADIUS(CM)                  ',
     .      '                   BZ, BT                     ')
          CALL BOOKPL(R, CPLOTN, NR, 1)
          CALL BOOKPL(R, CPLOTH, NR, -1)
          CALL ENDPLT
  100          CONTINUE
C
C       OUTPUT PROFILES
          WRITE(6,1000) CLOCK, DT, NSTEP
          WRITE(6,2000)
          WRITE(6,2010)(I,R(I),BZ(I),DEN(I,1),TEM(I,1),
     .      TEM(I,2),PTOT(I),BPRE(I),CONST(I),BT(I),I=1,NR)
          WRITE(6,2060)
          WRITE(6,2070)(I,R(I),JZ(I),JT(I),DELV(I),DELVZ(I),RESIST(I),
     .      I=1,NR)
C       LINER CALCULATIONS
          IF( .NOT. LBOOK) GO TO 530
          IF (LDSP) GO TO 530
C
C       VOLUME COMPRESSION RATIO
          DO 700 I=1,NR
          W(I) = ROLD(I)*DROLD(I)/(R(I)*DR(I))
  700     CONTINUE
C
```

```
*** MEMBER OUTPUT

205* C       THEORETICAL PROFILES                                              0205000
206*           DO 701 I=1,NR                                                   0206000
207*           DENT(I) = DENOLD(I)*W(I)                                        0207000
208*           TEMT(I,1) = TEMOLD(I,1)*W(I)**0.6667                            0208000
209*           TEMT(I,2) = TEMOLD(I,2)*W(I)**0.6667                            0209000
210*           PTOTT(I) = PTOTO(I) *W(I)**1.66667                              0210000
211*           BZT(I) = BZOLD(I) *W(I)                                         0211000
212*  701     CONTINUE                                                         0212000
213*           TOTPLN = 0.0                                                    0213000
214*           TOTPN = 0.0                                                     0214000
215*           DO 702 I=1,NR                                                   0215000
216* C                                                                         0216000
217* C      TOTAL PLASMA ENERGY DENSITY(NUMERICAL)                             0217000
218*           TOTPLN = TOTPLN + R(I)*DR(I)*PTOT(I)*1.5                        0218000
219* C                                                                         0219000
220* C      TOTAL ENERGY DENSITY OF B AND P (NUMERICAL)                        0220000
221*           TOTPN = TOTPN + (BZ(I)*BZ(I)/PI8 + 1.5*PTOT(I))                 0221000
222*     .       *DR(I)*R(I)                                                   0222000
223*  702     CONTINUE                                                         0223000
224* C                                                                         0224000
225* C      ENERGY DENSITY RATIO(NUMERICAL)                                    0225000
226*           ERATN = TOTPLN/TOTPN                                            0226000
227*           TOTPL = 0.0                                                     0227000
228*           TOTP = 0.0                                                      0228000
229*           DO 703 I=1,NR                                                   0229000
230* C                                                                         0230000
231* C      TOTAL PLASMA ENERGY DENSITY(ANALYTICAL)                            0231000
232*           TOTPL = TOTPL + 1.5*DR(I)*R(I)*LOCBET(I)*W(I)**0.6667           0232000
233* C                                                                         0233000
234* C      TOTAL PLASMA AND MAGNETIC ENERGY DENSITY(ANALYTICAL)               0234000
235*           TOTP = TOTP + ((1.0 - LOCBET(I))*W(I) + 1.5*LOCBET(I)           0235000
236*     .       *W(I)**0.66667)*DR(I)*R(I)                                    0236000
237*  703     CONTINUE                                                         0237000
238* C                                                                         0238000
239* C      ENERGY DENSITY RATIO(ANALYTICAL)                                   0239000
240*           ERAT = TOTPL/TOTP                                               0240000
241* C                                                                         0241000
242* C      OUTPUT DIFFERENT QUANTITIES                                        0242000
243*           WRITE(6,7777) ERAT, ERATN                                       0243000
244* 7777      FORMAT(1H1,1X,'RATIO OF PLASMA TO TOTAL ENERGY:',8X,            0244000
245*     .       'THEORY',2X,1PG12.4,8X,'EXP',2X,1PG12.4///)                   0245000
246*           WRITE(6,7778)                                                   0246000
247* 7778      FORMAT(1X,'GRID',7X,'R',6X,'N',12X,'TE',10X,'TI',10X,           0247000
248*     .       'PL-TOT',10X,'B',8X,'COM-RAT'//)                            0248000
249*           WRITE(6,7779) (I,R(I),DENT(I),TEMT(I,1),TEMT(I,2),              0249000
250*     .       PTOTT(I),BZT(I),W(I),I=1,NR)                                0250000
251* 7779      FORMAT(1X, I3, 7E12.4)                                          0251000
252*  530      CONTINUE                                                        0252000
253* C                                                                         0253000
254* C      OUTPUT ALL CONSERVATIVE QUANTITIES                                 0254000
255*           TOTPLN = 0.0                                                    0255000
```

*** MEMBER OUTPUT

```
256*            TOTBE = 0.0                                                     0256000
257*            BTTOT = 0.0                                                     0257000
258*            NTOTAL = 0.0                                                    0258000
259*            BFLUX = 0.0                                                     0259000
260*            VOL = 0.0                                                       0260000
261*            DO 610 I=1,NR                                                   0261000
262*            DV(I) = PI2*R(I)*DR(I)                                          0262000
263*            VOL = VOL + DV(I)                                               0263000
264*            BTTOT = BTTOT + BT(I)*DR(I)                                     0264000
265*            ITOTZ = ITOTZ + JZ(I)*DV(I)                                     0265000
266*            ITOTT = ITOTT + DR(I)*JT(I)                                     0266000
267*            BFLUX = BFLUX + DV(I)*BZ(I)                                     0267000
268*            NTOTAL = NTOTAL + 2.0*DV(I)*DEN(I,1)                            0268000
269*            TOTBE = TOTBE + (BZ(I)**2 + BT(I)**2)*DV(I)/PI8                 0269000
270*            TOTPLN = TOTPLN + PTOT(I)*1.5*DV(I)                             0270000
271*  610       CONTINUE                                                        0271000
272*            TOTE = TOTBE + TOTPLN                                           0272000
273*            DIAM = 0.0                                                      0273000
274*            DO 640 I=1,23                                                   0274000
275*  640       DIAM = DIAM + PI2*R(I)*DR(I)*BZ(I)                              0275000
276*            DIAM = DIAM/DIAM0                                               0276000
277*            BTRAT = BT(23)/BTOLD                                            0277000
278*            ERATN = TOTPLN/TOTE                                             0278000
279*            WALLE = (COLD + KNEW)/2.0*PI*(WOLD**2 - RWALL**2) + WALLE       0279000
280*            COLD = KNEW                                                     0280000
281*            WOLD = RWALL                                                    0281000
282*            SYSE = TOTE - WALLE                                             0282000
283*            WRITE(6,620)                                                    0283000
284*  620       FORMAT(/1X,'NTOTAL          TOTE        B-ENER        P-ENER    0284000
285*           .ITOTZ      ITOTT      RZTOTT       BTTOT     DIAM    BTRAT')   0285000
286*            WRITE(6,630)(NTOTAL,TOTE,TOTBE,TOTPLN,ITOTZ,ITOTT,BFLUX,        0286000
287*           .  BTTOT, DIAM, BTRAT)                                           0287000
288*  630       FORMAT(1X,9(E12.4,1X))                                          0288000
289* C                                                                          0289000
290* C                                                                          0290000
291*            IF(.NOT. LTURB) GO TO 600                                       0291000
292*            WRITE(6,2040)                                                   0292000
293*            WRITE(6,2050)(I,RG(I), VTG(I,1), VTG(I,2), VCRIT(I), VD(I),     0293000
294*           .  DELVG(I), VEB(I), ANDV(I), ANDHT(I), I=1,NRP)                 0294000
295* 2040      FORMAT(1H1,1X,'GRID',5X,'RG',10X,'VTE',8X,'VTI',8X,              0295000
296*           .  'VCRIT',8X,'VD',8X,'-VDIAM',8X,'VEB',8X,'TURB-V',8X,          0296000
297*           .  'TURB-Q')                                                     0297000
298* 2050      FORMAT(1X,I3,9E12.4)                                            0298000
299*  600       CONTINUE                                                        0299000
300*            RETURN                                                          0300000
301*          END                                                               0301000
```

```
*** MEMBER RSPRNT

      1*          SUBROUTINE RSPRNT
      2*          IMPLICIT REAL*4 (A-Z)
      3*          PARAMETER NRP=29, NR=28, NS=2
      4*          COMMON/CON2/ GAM, KNEW
      5*          COMMON/CON1/ PI, PI2, PI4, PI6, PI8, BK, C
      6*          COMMON/GRID/ RG(NRP), R(NR), C1(NRP), C2(NRP), DR(NRP)
      7*          COMMON/FLUXG/ MASS(NRP,NS), S(NRP,NS), BZFLX(NRP),BTFLX(NRP)
      8*          COMMON/MTG/ BZG(NRP), DENG(NRP,NS), TEMG(NRP,NS), PG(NRP,NS)
      9*         .   , PTOTG(NRP), BTG(NRP)
     10*          NAMELIST/NL/ GAM, KNEW, RG, R, C1, C2, DR, MASS, S,
     11*         .   BZFLX, BTFLX, BZG,BTG,DENG,TEMG,PG,PTOTG
     12*          WRITE(6,NL)
     13*          RETURN
     14*          END
```

```
*** MEMBER STEP

      1*        SUBROUTINE STEP( FRAC, TFACT, RRIGHT, RMIN, TMID, DT, NPRINT,
      2*     .     NSTOP, NPLOT)
      3* C
      4* C     STEP CONTROLS THE FLOW OF SUBROUTINES IN TIME DT
      5* C     NPRINT = OUTPUT SKIPS          , LPRINT = OUTPUT KEY
      6* C     LDSP = KEY FOR TURNING ON DISSIPATION
      7* C        LWALL = KEY FOR TURNING ON WALL MOTION
      8* C
      9*           IMPLICIT REAL*4 ( A-Z)
     10* *         VDIM
     11*           COMMON/SWITCH/ LPRINT, LDSP, LWALL, LPLOT, LBOOK
     12*           INTEGER NSTEP, NPRINT, NSTOP, NPLOT
     13*           LOGICAL LPRINT, LDSP, LWALL, LPLOT,  LBOOK
     14* C
     15*           CLOCK = 0.0
     16*           NSTEP = 0
     17*  100      NSTEP = NSTEP + 1
     18*           CALL FLUX
     19*           CLOCK = CLOCK + DT
     20*           IF (LWALL) CALL WALL (CLOCK, RRIGHT, RMIN, TMID)
     21*           CALL LAGRAG
     22*           LPRINT = .FALSE.
     23*           CALL OUTPUT(CLOCK, DT, NSTEP, NPLOT)
     24*           IF(LDSP) CALL FLOW( DT, TFACT, FRAC, NSTEP)
     25*           LPRINT = MOD(NSTEP,NPRINT) .EQ. 0
     26*           CALL OUTPUT(CLOCK, DT, NSTEP, NPLOT)
     27*           IF (NSTEP .LT. NSTOP) GO TO 100
     28*           RETURN
     29*           END
```

```
*** MEMBER TINCON

      SUBROUTINE TINCON(DRG, ETA, R0, ALFA, DEN0, BZ0, B1, T0)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     TINCON COMPUTES INITIAL CONDITIONS FOR THETA PINCH SETUP
C     DRG    = GRID SPACING          , ETA    = TI/TE
C     R0     = LOCATION OF SHEATH    , ALFA   = SHEATH THICKNESS
C     DEN0   = MAX DENSITY           , BZ0    = MAX BZ
C     TEMG   = TEMPERATURE AT GRID POINT
C     TEM    = TEMPERATURE AT MID POINT
C     DEN    = DENSITY               , P      = PARTIAL PRESSURE
C     PTOT   = TOTAL PLASMA PRESSURE, RHO    = MASS DENSITY
C     GAM    = ADIABATIC GAMMA       , KNEW   = TOTAL PRESSURE CONSTANT
C     ELECTRON IS FIRST SPECIE       , ION IS SECOND SPECIE
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
         IMPLICIT REAL*4 (A -Z)
#        VDIM
         COMMON/CON1/ PI, PI2, PI4, PI6, PI8, BK, C
         COMMON/GRID/ RG(NRP), R(NR), C1(NRP), C2(NRP), DR(NRP)
         COMMON/SPECIE/ AM(NS), Q(NS)
         COMMON/MTM/ BZ(NR), DEN(NR,NS), TEM(NR,NS), P(NR,NS), PTOT(NR)
         COMMON/MTG/ BZG(NRP), DENG(NRP,NS), TEMG(NRP,NS), PG(NRP,NS),
                     PTOTG(NRP)
         COMMON/CON2/ GAM, KNEW
         COMMON/OUT2/ BASE
C
         REAL*4 RHO(NR,NS)
         INTEGER I, J
 1000    FORMAT(//4X,'SPECIE ',I2,9X,'Q= ',1PG12.4,9X,'M= ',1PG12.4)
 2000    FORMAT(//4X,'INITIAL BETA = ',1PG12.4)
C
C
         BASE=0.0
         DO 100 I=1, NRP
 100     RG(I) = FLOAT(I-1) * DRG
         DO 200 I=1, NRP
 200     R(I) = (RG(I) + RG(I+1))*0.5
C
C     INITIAL PROFILES AT GRID POINTS
         C3 = 1.0/((1.0 + ETA)*BK)
         C4 = C3/PI8
         BMAX2 = BZ0*BZ0/PI8
         DO 300 I= 1, NRP
         ARG = (RG(I)**2 - R0**2)/ALFA**2
         TANA = TANH(ARG)
         COSA = COSH(ARG)
         DENG(I,1) = DEN0*(1.0 - TANA) /2.0
         DENG(I,2) = DENG(I,1)
         TEMG(I,1) = T0*(1.0 - TANA)/2.0
         TEMG(I,2) = TEMG(I,1)*ETA
         BZG(I) = SQRT(PI8*(BMAX2 - DENG(I,1)*BK*TEMG(I,1)
```

```
*** MEMBER TINCON

52*                    *(1.0 + ETA)))
53*  300          CONTINUE
54*               DO 400 I=1, NRP
55*               DENK = DENG(I,1)*BK
56*               PG(I,1) = DENK*TEMG(I,1)
57*               PG(I,2) = DENK*TEMG(I,2)
58*  400          PTOTG(I) = PG(I,1) + PG(I,2)
59* C
60* C         INITIAL PROFILES AT MID POINTS
61*               DO 500 I=1,NR
62*               DEN(I,1) = (DENG(I,1) + DENG(I+1,1))/2.0
63*               DEN(I,2) = DEN(I,1)
64*               BZ(I) = (BZG(I+1) + BZG(I))/2.0
65*
66*               TEM(I,1) = (TEMG(I+1,1) + TEMG(I,1))/2.0
67*               TEM(I,2) = (TEMG(I+1,2) + TEMG(I,2))/2.0
68*               DENK = DEN(I,1)*BK
69*               P(I,1) = DENK*TEM(I,1)
70*               P(I,2) = DENK*TEM(I,2)
71*               PTOT(I) = P(I,1) +P(I,2)
72*               RHO(I,1) = DEN(I,1)*AM(1)
73*               RHO(I,2) = DEN(I,2)*AM(2)
74*  500          CONTINUE
75* C
76* C         INITIAL AKNEW(PLASMA AND MAGNETIC PRESSURE)
77*               KNEW = PTOT(NR) + BZ(NR)**2/PI8
78*               BETA = PTOT(1)/(BZ(NR)**2/PI8)
79*               DO 550 I=1,NS
80*               WRITE(6,1000) I, Q(I), AM(I)
81*  550          CONTINUE
82*               WRITE(6,2000) BETA
83*               RETURN
84*               END
```

```
*** MEMBER TRID

      1*        SUBROUTINE TRID(U, A, B, C, D)
      2* C
      3* C      TRID SOLVES TRIDIAGONAL
      4* C       A*U(I-1) + B*U(I) + C*U(I+1) = D
      5* C
      6*           IMPLICIT REAL*4 (A-Z)
      7* *           VDIM
      8*           INTEGER I, M, J
      9*           REAL U(NR), A(NR), B(NR), C(NR), D(NR), X(NR), Y(NR)
     10*           M = NR- 1
     11*           X(NR) = 0.0
     12*           Y(NR) = 0.0
     13*           X(M) = -A(NR)/B(NR)
     14*           Y(M) = D(NR)/B(NR)
     15*           DO 100 I=2,M
     16*           J = NRP -I
     17*           X(NR-I) = -A(J)/(C(J)*X(J) + B(J))
     18*           Y(NR-I) = (D(J) - C(J)*Y(J))/(C(J)*X(J) + B(J))
     19*  100      CONTINUE
     20*           U(1) = (D(1) - C(1)*Y(1))/(B(1) + C(1)*X(1))
     21*           DO 200 I=1,M
     22*           U(I+1) = X(I)*U(I) + Y(I)
     23*  200      CONTINUE
     24*           RETURN
     25*           END
```

```
*** MEMBER TSTEP

      1*          SUBROUTINE TSTEP(DT, TFACT, BDIF, DENG)
      2* C
      3*          RETURN
      4*          END
```

```
*** MEMBER TURB

      1*            SUBROUTINE TURB(DELVG,DELVZ,WCG,GNUPRP,GNUPAR,BG)                0001000
      2*           IMPLICIT REAL*8(A-Z)                                              0002000
      3*           PARAMETER NRP=29, NS=2, NP=NRP-1                                  0003000
      4*           COMMON/CON1/ PI, PI2, PI4, PI6, PI8, PK, C                        0004000
      5*           COMMON/SPECIE/ AM(NS), Q(NS)                                      0005000
      6*           COMMON/MTG/ BZG(NRP), DENG(NRP,NS), TEMG(NRP,NS), PG(NRP,NS),     0006000
      7*     .PTOTG(NRP),BTG(NRP)                                                    0007000
      8*           REAL*8 PAT1(NRP), SATE(NRP), ALFG(NRP), SPG(NRP,NS),              0008000
      9*     .DELVZ(NRP), GNUPRP(NRP),GNUPAR(NRP), WLHG(NRP),                        0009000
     10*     .WPG(NRP,NS),WCG(NRP,NS),DELVG(NRP),PG(NRP),VTG(NRP,NS),                0010000
     11*     .VCRIT(NRP)                                                             0011000
     12*           INTEGER I                                                         0012000
     13* C                                                                           0013000
     14* C                                                                           0014000
     15*           DO 100 I=1,NRP                                                    0015000
     16*           VTG(I,1) = 4.19E7*SQRT(TEMG(I,1))*1.414                           0016000
     17*           VTG(I,2) = 9.79E5*SQRT(TEMG(I,2))                                 0017000
     18*           WPG(I,1) = 5.64E4*SQRT(DENG(I,1))                                 0018000
     19*           WPG(I,2) = 1.32E3*1.414*SQRT(DENG(I,1))                           0019000
     20*           PAT1(I) = 1.0 + (WPG(I,1)/WCG(I,1))**2                            0020000
     21*           WLHG(I) = WPG(I,2)/SQRT(PAT1(I))                                  0021000
     22*  100      CONTINUE                                                          0022000
     23*           RTMASS = AM(1)/AM(2)                                              0023000
     24* C                                                                           0024000
     25* C     POST IMPLOSION DRIFT TRANSPORT                                        0025000
     26* C                                                                           0026000
     27*           DO 200 I=1,NRP                                                    0027000
     28* C                                                                           0028000
     29* C     CALCULATE CRITICAL VELOCITY                                           0029000
     30*           VCRIT(I) = 1.414*WCG(I,2)/WLHG(I)*VTG(I,2)                        0030000
     31* C                                                                           0031000
     32* C       CALCULATE PERP AND PARR CURRENT VELOCITIES                          0032000
     33*           SSIN=BTG(I)/BG(I)                                                 0033000
     34*           CCOS=BZG(I)/BG(I)                                                 0034000
     35*             VZG=0.5*(DELVZ(I-1)+DELVZ(I))                                   0035000
     36*           VDPRP=CCOS*DELVG(I)-SSIN*VZG                                      0036000
     37*           VDPAR=SSIN*DELVG(I)+CCOS*VZG                                      0037000
     38* C                                                                           0038000
     39* C     LOWER HYBRID TRANSPORT                                                0039000
     40*           IF (ABS(VDPRP) .GT. VTG(I,2) ) GO TO 200                          0040000
     41*           IF (ABS(VDPRP) .LT. VCRIT(I) ) GO TO 200                          0041000
     42* C                                                                           0042000
     43* C     SATURATION LEVEL IS ALWAYS USED                                       0043000
     44*           SATE(I) = DENG(I,1)*AM(1)*VDPRP**2/(4.0*PAT1(I))                  0044000
     45*           GNUPRP(I)=SQRT(PI/2.0)*PAT1(I)*WLHG(I)*2.0*SATE(I)/               0045000
     46*      .    (DENG(I,1)*AM(1)*VTG(I,2)**2)                                     0046000
     47*  200      CONTINUE                                                          0047000
     48*           RETURN                                                            0048000
     49*           END                                                               0049000
```

```
*** MEMBER WALL

      1*        SUBROUTINE WALL (CLOCK, RRIGHT, RMIN, TMID)
      2* C
      3* C      WALL COMPUTES THE WALL DISPLACEMENT IN TIME T)
      4* C
      5*           IMPLICIT REAL*4 (A-Z)
      6*           COMMON/RWAL/ RWALL
      7*           RWALL = RMIN + (RRIGHT - RMIN)*ABS(CLOCK - TMID)/TMID
      8*           RETURN
      9*        END
```

Appendix C
Initial and Final Conditions for Test Problems

ADIABATIC COMPRESSION
Initial Conditions

```
&INPUT
GAM = 1.6666670,      DRG = 0.25000000,     ETA = 1.0000000,    RO = 30.000000,      ALFA = 25.000000,
DENO = 0.00000000,,   BZO = 2400.0000,      TFACT = .0000000,   B1 = 100.0000,       RRIGHT = 7.0000000,
RMIN = 1.0000000,     TMID = 0.59999991E-03, TO = 10.0000000,   DT = 0.59999993E-05, RWALL = 7.0000000,
NPRINT = 25,          NSTOP = 200,          BTO = -2000.0000,   TLBASE = 0.00000000, TRBASE = 0.00000000,
NFRAC = 0.10000000E-03, TFRAC = 0.80000001, NPLOT = 5,
&END,
&KNOB,
RJP = 2.0000000,      DRJZ = 1.0000000,     DRJT = 1.5000000,
LPRINT = T,           LDSP = F,             LWALL = T,          LPLOT = T,           LBOOK = T,
LTURB = F,            LDUMP = F,
&END,
```

TIME = 0.00 TIME-STEP = 0.00 STEP = 0

GRID	R	B	N	TE	TI	PTOTAL	PP	CONSTANT	BT
1	0.1250E 00	-0.3742E 04	0.3250E 16	0.9802E 00	0.9802E 00	0.1021E 05	0.5571E 06	0.5673E 06	-0.3716E 01
2	0.3750E 00	-0.3646E 04	0.4106E 16	0.2903E 01	0.2903E 01	0.3819E 05	0.5291E 06	0.5673E 06	-0.1895E 02
3	0.6250E 00	-0.3484E 04	0.5560E 16	0.4714E 01	0.4714E 01	0.8309E 05	0.4833E 06	0.5673E 06	-0.5680E 02
4	0.8750E 00	-0.3231E 04	0.7425E 16	0.6344E 01	0.6344E 01	0.1507E 06	0.4163E 06	0.5673E 06	-0.1345E 03
5	0.1125E 01	-0.2864E 04	0.9556E 16	0.7730E 01	0.7730E 01	0.2367E 06	0.3306E 06	0.5673E 06	-0.2719E 03
6	0.1375E 01	-0.2377E 04	0.1167E 17	0.8819E 01	0.8819E 01	0.3297E 06	0.2376E 06	0.5673E 06	-0.4832E 03
7	0.1625E 01	-0.1779E 04	0.1334E 17	0.9569E 01	0.9569E 01	0.4022E 06	0.1581E 06	0.5673E 06	-0.7666E 03
8	0.1875E 01	-0.1103E 04	0.1418E 17	0.9952E 01	0.9952E 01	0.4521E 06	0.1152E 06	0.5673E 06	-0.1098E 04
9	0.2125E 01	-0.3992E 03	0.1374E 17	0.9952E 01	0.9952E 01	0.4405E 06	0.1228E 06	0.5673E 06	-0.1433E 04
10	0.2375E 01	0.2765E 03	0.1270E 17	0.9569E 01	0.9569E 01	0.3896E 06	0.1777E 06	0.5673E 06	-0.1721E 04
11	0.2625E 01	0.8740E 03	0.1063E 17	0.8819E 01	0.8819E 01	0.3061E 06	0.2612E 06	0.5673E 06	-0.1925E 04
12	0.2875E 01	0.1362E 04	0.8774E 16	0.7730E 01	0.7730E 01	0.2178E 06	0.3495E 06	0.5673E 06	-0.2031E 04
13	0.3125E 01	0.1728E 04	0.7008E 16	0.6344E 01	0.6344E 01	0.1425E 06	0.4248E 06	0.5673E 06	-0.2007E 04
14	0.3375E 01	0.1982E 04	0.5776E 16	0.4714E 01	0.4714E 01	0.8756E 05	0.4797E 06	0.5673E 06	-0.2000E 04
15	0.3625E 01	0.2144E 04	0.5620E 16	0.2903E 01	0.2903E 01	0.5228E 05	0.5150E 06	0.5673E 06	-0.1915E 04
16	0.3875E 01	0.2237E 04	0.1015E 17	0.9802E 00	0.9802E 00	0.3188E 05	0.5354E 06	0.5673E 06	-0.1816E 04
17	0.4125E 01	0.2291E 04	0.1318E 17	0.5000E 00	0.5000E 00	0.2112E 05	0.5462E 06	0.5673E 06	-0.1715E 04
18	0.4375E 01	0.2317E 04	0.9916E 16	0.5000E 00	0.5000E 00	0.1589E 05	0.5514E 06	0.5673E 06	-0.1621E 04
19	0.4625E 01	0.2329E 04	0.8436E 16	0.5000E 00	0.5000E 00	0.1353E 05	0.5537E 06	0.5673E 06	-0.1534E 04
20	0.4875E 01	0.2334E 04	0.7831E 16	0.5000E 00	0.5000E 00	0.1255E 05	0.5547E 06	0.5673E 06	-0.1456E 04
21	0.5125E 01	0.2336E 04	0.7591E 16	0.5000E 00	0.5000E 00	0.1216E 05	0.5551E 06	0.5673E 06	-0.1385E 04
22	0.5375E 01	0.2336E 04	0.7503E 16	0.5000E 00	0.5000E 00	0.1202E 05	0.5553E 06	0.5673E 06	-0.1320E 04
23	0.5625E 01	0.2337E 04	0.7472E 16	0.5000E 00	0.5000E 00	0.1197E 05	0.5553E 06	0.5673E 06	-0.1261E 04
24	0.5875E 01	0.2337E 04	0.7461E 16	0.5000E 00	0.5000E 00	0.1195E 05	0.5553E 06	0.5673E 06	-0.1208E 04
25	0.6125E 01	0.2337E 04	0.7456E 16	0.5000E 00	0.5000E 00	0.1195E 05	0.5553E 06	0.5673E 06	-0.1158E 04
26	0.6375E 01	0.2337E 04	0.7453E 16	0.5000E 00	0.5000E 00	0.1194E 05	0.5553E 06	0.5673E 06	-0.1113E 04
27	0.6625E 01	0.2337E 04	0.7452E 16	0.5000E 00	0.5000E 00	0.1191E 05	0.5553E 06	0.5673E 06	-0.1071E 04
28	0.6875E 01	0.2368E 04	0.3754E 16	0.5000E 00	0.5000E 00	0.6015E 04	0.5613E 06	0.5673E 06	-0.1032E 04

GRID	R	JZ	JT	DELV	DELVZ	RESIST
1	0.1250E 00	-0.2169E 12	-0.7808E 12	0.5003E 06	0.1396E 06	0.1744E-12
2	0.3750E 00	-0.4103E 12	-0.1228E 13	0.6229E 06	0.2081E 06	0.3422E-13
3	0.6250E 00	-0.8069E 12	-0.1985E 13	0.7434E 06	0.3022E 06	0.1653E-13
4	0.8750E 00	-0.1434E 13	-0.2961E 13	0.8303E 06	0.4022E 06	0.1059E-13
5	0.1125E 01	-0.2281E 13	-0.4177E 13	0.8883E 06	0.4969E 06	0.7874E-14
6	0.1375E 01	-0.3232E 13	-0.5183E 13	0.9249E 06	0.5769E 06	0.6462E-14
7	0.1625E 01	-0.4078E 13	-0.6081E 13	0.9408E 06	0.6304E 06	0.5717E-14
8	0.1875E 01	-0.4579E 13	-0.6587E 13	0.9674E 06	0.6724E 06	0.5390E-14
9	0.2125E 01	-0.4572E 13	-0.6587E 13	0.9840E 06	0.6830E 06	0.5390E-14
10	0.2375E 01	-0.4061E 13	-0.6081E 13	0.9956E 06	0.6655E 06	0.5717E-14
11	0.2625E 01	-0.3266E 13	-0.5183E 13	0.9961E 06	0.6163E 06	0.6461E-14
12	0.2875E 01	-0.2250E 13	-0.4077E 13	0.9654E 06	0.5328E 06	0.7874E-14
13	0.3125E 01	-0.1403E 13	-0.2961E 13	0.8797E 06	0.4169E 06	0.1059E-13
14	0.3375E 01	-0.7768E 12	-0.1985E 13	0.7130E 06	0.2790E 06	0.1653E-13
15	0.3625E 01	-0.3809E 12	-0.1228E 13	0.4550E 06	0.1411E 06	0.3422E-13
16	0.3875E 01	-0.1648E 12	-0.7016E 12	0.1439E 06	0.3381E 05	0.1744E-12
17	0.4125E 01	-0.6229E 11	-0.3698E 12	0.5841E 05	0.9839E 04	0.4787E-12
18	0.4375E 01	-0.1987E 11	-0.1798E 12	0.3776E 05	0.4171E 04	0.4787E-12
19	0.4625E 01	-0.4662E 10	-0.8073E 11	0.1900E 05	0.1149E 04	0.4787E-12
20	0.4875E 01	-0.1530E 08	-0.3343E 11	0.8889E 04	0.4069E 01	0.4787E-12
21	0.5125E 01	0.1072E 10	-0.1277E 11	0.3512E 04	-0.2794E 03	0.4787E-12
22	0.5375E 01	0.1180E 10	-0.4495E 10	0.1247E 04	-0.3274E 03	0.4787E-12
23	0.5625E 01	0.1041E 10	-0.1406E 10	0.4080E 03	-0.2901E 03	0.4787E-12
24	0.5875E 01	0.8952E 09	-0.4383E 09	0.1223E 03	-0.2498E 03	0.4787E-12
25	0.6125E 01	0.7491E 09	-0.1096E 09	0.3600E 02	-0.2092E 03	0.4787E-12
26	0.6375E 01	0.6436E 09	-0.1632E 08	0.4539E 01	-0.1798E 03	0.4787E-12
27	0.6625E 01	0.5714E 09	-0.1510E 12	0.6218E 05	-0.1605E 03	0.4787E-12
28	0.6875E 01	0.3521E 10	-0.1510E 12	0.8471E 05	-0.1953E 04	0.4787E-12

NTOTAL	TOTE	[illegible]	[illegible]	ITOTZ	ITOTT	HZTOTT	BTTOT	DIAM	BTHAT
0.2545E 17	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.2676E 06	-0.8378E 04	0.1000E 01	
0.1000E 01									

TIME = 6.00E-04 TIME-STEP = 6.00E-06 STEP = 100

GRID	R	Q	N	TE	TI	PTOTAL	PE	CONSTANT	BT
1	0.2266E-01	-0.1139E 06	0.9888E 17	0.9552E 01	0.9553E 01	0.3027E 07	0.5158E 09	0.5188E 09	-0.2056E 02
2	0.6769E-01	-0.1123E 06	0.1271E 18	0.2862E 02	0.2862E 02	0.1166E 08	0.5071E 09	0.5188E 09	-0.1659E 03
3	0.1119E 00	-0.1112E 06	0.1774E 18	0.4742E 02	0.4742E 02	0.2696E 08	0.4918E 09	0.5188E 09	-0.3298E 03
4	0.1549E 00	-0.1082E 06	0.2488E 18	0.6593E 02	0.6593E 02	0.5256E 08	0.4662E 09	0.5188E 09	-0.7977E 03
5	0.1958E 00	-0.1034E 06	0.3442E 18	0.9443E 02	0.9443E 02	0.9332E 08	0.4255E 09	0.5188E 09	-0.1708E 04
6	0.2340E 00	-0.9549E 05	0.4687E 18	0.1035E 03	0.1035E 03	0.1554E 09	0.3634E 09	0.5188E 09	-0.3304E 04
7	0.2686E 00	-0.8261E 05	0.6197E 18	0.1236E 03	0.1236E 03	0.2455E 09	0.2733E 09	0.5188E 09	-0.5885E 04
8	0.2990E 00	-0.6122E 05	0.7868E 18	0.1448E 03	0.1448E 03	0.3650E 09	0.1538E 09	0.5188E 09	-0.9715E 04
9	0.3255E 00	-0.2691E 05	0.9219E 18	0.1628E 03	0.1628E 03	0.4806E 09	0.3800E 08	0.5188E 09	-0.1451E 05
10	0.3496E 00	0.1993E 05	0.9157E 18	0.1657E 03	0.1657E 03	0.4863E 09	0.3248E 08	0.5188E 09	-0.1826E 05
11	0.3741E 00	0.6027E 05	0.7467E 18	0.1483E 03	0.1483E 03	0.3547E 09	0.1641E 09	0.5188E 09	-0.1891E 05
12	0.4012E 00	0.8431E 05	0.5476E 18	0.1214E 03	0.1214E 03	0.2131E 09	0.3057E 09	0.5188E 09	-0.1765E 05
13	0.4316E 00	0.9784E 05	0.3968E 18	0.9353E 02	0.9353E 02	0.1189E 09	0.3999E 09	0.5188E 09	-0.1601E 05
14	0.4649E 00	0.1046E 06	0.3059E 18	0.6632E 02	0.6632E 02	0.6500E 08	0.4538E 09	0.5188E 09	-0.1453E 05
15	0.4999E 00	0.1081E 06	0.2832E 18	0.3760E 02	0.3760E 02	0.3594E 08	0.4829E 09	0.5188E 09	-0.1331E 05
16	0.5363E 00	0.1096E 06	0.4977E 18	0.1313E 02	0.1313E 02	0.2094E 08	0.4979E 09	0.5188E 09	-0.1232E 05
17	0.5734E 00	0.1104E 06	0.6365E 18	0.6630E 01	0.6630E 01	0.1352E 08	0.5053E 09	0.5188E 09	-0.1151E 05
18	0.6107E 00	0.1110E 06	0.4752E 18	0.6597E 01	0.6597E 01	0.1004E 08	0.5088E 09	0.5188E 09	-0.1084E 05
19	0.6481E 00	0.1112E 06	0.4033E 18	0.6581E 01	0.6581E 01	0.8505E 07	0.5103E 09	0.5188E 09	-0.1026E 05
20	0.6854E 00	0.1113E 06	0.3734E 18	0.6574E 01	0.6574E 01	0.7865E 07	0.5109E 09	0.5188E 09	-0.9757E 04
21	0.7226E 00	0.1113E 06	0.3617E 18	0.6571E 01	0.6571E 01	0.7616E 07	0.5112E 09	0.5188E 09	-0.9302E 04
22	0.7598E 00	0.1113E 06	0.3574E 18	0.6570E 01	0.6570E 01	0.7525E 07	0.5113E 09	0.5188E 09	-0.8889E 04
23	0.7969E 00	0.1113E 06	0.3559E 18	0.6570E 01	0.6570E 01	0.7492E 07	0.5113E 09	0.5188E 09	-0.8511E 04
24	0.8339E 00	0.1113E 06	0.3553E 18	0.6570E 01	0.6570E 01	0.7480E 07	0.5113E 09	0.5188E 09	-0.8164E 04
25	0.8709E 00	0.1113E 06	0.3551E 18	0.6569E 01	0.6569E 01	0.7475E 07	0.5113E 09	0.5188E 09	-0.7844E 04
26	0.9077E 00	0.1113E 06	0.3550E 18	0.6569E 01	0.6569E 01	0.7473E 07	0.5113E 09	0.5188E 09	-0.7547E 04
27	0.9446E 00	0.1113E 06	0.3549E 18	0.6569E 01	0.6569E 01	0.7471E 07	0.5113E 09	0.5188E 09	-0.7271E 04
28	0.9816E 00	0.1117E 06	0.1771E 18	0.6527E 01	0.6527E 01	0.3704E 07	0.5151E 09	0.5188E 09	-0.6989E 04

GRID	R	JZ	JT	DELV	DELVZ	RESIST
1	0.2266E-01	-0.6686E 13	-0.4294E 14	0.2011E 06	0.1498E 06	0.5732E-14
2	0.6769E-01	-0.1311E 14	-0.7167E 14	0.1174E 07	0.2147E 06	0.1105E-14
3	0.1119E 00	-0.2739E 14	-0.1279E 15	0.1501E 07	0.3215E 06	0.5182E-15
4	0.1549E 00	-0.5379E 14	-0.2231E 15	0.1867E 07	0.4592E 06	0.3161E-15
5	0.1958E 00	-0.9963E 14	-0.3884E 15	0.2344E 07	0.6013E 06	0.2182E-15
6	0.2340E 00	-0.1758E 15	-0.6918E 15	0.3073E 07	0.7808E 06	0.1608E-15
7	0.2686E 00	-0.2955E 15	-0.1205E 16	0.4316E 07	0.9928E 06	0.1231E-15
8	0.2990E 00	-0.4479E 15	-0.2010E 16	0.6378E 07	0.1184E 07	0.9714E-16
9	0.3255E 00	-0.5071E 15	-0.3863E 16	0.8725E 07	0.1145E 07	0.8150E-16
10	0.3496E 00	-0.3367E 15	-0.4258E 16	0.9682E 07	0.7656E 06	0.7931E-16
11	0.3741E 00	-0.9371E 14	-0.3005E 16	0.8492E 07	0.2613E 06	0.9374E-16
12	0.4012E 00	0.1552E 14	-0.1593E 16	0.6055E 07	-0.5901E 05	0.1265E-15
13	0.4316E 00	0.2858E 14	-0.7549E 15	0.3961E 07	-0.1500E 06	0.1871E-15
14	0.4648E 00	0.1940E 14	-0.3600E 15	0.2450E 07	-0.1320E 06	0.3139E-15
15	0.4999E 00	0.9901E 13	-0.1765E 15	0.1293E 07	-0.7278E 05	0.6790E-15
16	0.5363E 00	0.3350E 13	-0.9036E 14	0.3522E 06	-0.1003E 05	0.3557E-14
17	0.5734E 00	-0.5411E 12	-0.3978E 14	0.1301E 06	0.1770E 04	0.9912E-14
18	0.6107E 00	-0.2502E 13	-0.1792E 14	0.7853E 05	0.1096E 05	0.9929E-14
19	0.6481E 00	-0.3216E 13	-0.7556E 13	0.3911E 05	0.1600E 05	0.1002E-13
20	0.6854E 00	-0.3236E 13	-0.2916E 13	0.1626E 05	0.1800E 05	0.1004E-13
21	0.7226E 00	-0.2934E 13	-0.9744E 12	0.5609E 04	0.1699E 05	0.1005E-13
22	0.7599E 00	-0.2547E 13	-0.2311E 12	0.1358E 04	0.1084E 05	0.1005E-13
23	0.7969E 00	-0.2172E 13	[illegible]	[illegible]	0.1271E 05	0.1005E-13
24	0.8339E 00	-0.1840E 13	[illegible]	-0.4728E 03	0.1678E 05	0.1005E-13
25	0.8709E 00	-0.1561E 13	0.2296E 11	-0.5959E 03	0.9153E 04	0.1005E-13
26	0.9077E 00	-0.1332E 13	0.3665E 11	-0.4748E 03	0.7611E 04	0.1005E-13
27	0.9446E 00	0.9772E 12	-0.1367E 14	0.8136E 05	-0.5733E 04	0.1005E-13
28	0.9816E 00	0.1752E 13	-0.1391E 14	0.1035E 06	-0.2060E 05	0.1015E-13

SEEBIE EXPERIMENT
Initial Conditions

```
&INPUT
GAM = 1.6666670,        DRG = 0.25000000,       ETA = 1.0000000,        RO = 30.000000,         ALFA = 25.000000,
DENO = -0.00000000,     BZO = 2400.0000,        TFACT = 5.0000000,      B1 = 100.00000,         RRIGHT = 7.0000000,
RMIN = 1.0000000,       TMID = 0.99999990E-04,  TO = 10.0000000,        DT = 0.99999973E-08,    RWALL = 7.0000000,
NPRINT = 10,            NSTOP = 400,            BTO = -2000.0000,       TLBASE = 0.00000000,    TRBASE = 0.00000000,
NFRAC = 0.10000000E-03, TFRAC = 0.80000001,     NPLOT = 100,            RJP = 2.0000000,        DRJZ = 1.0000000,
DRJT = 1.5000000,
&END,
&KNOB,
LPRINT = T,             LDSP = T,               LWALL = F,              LPLOT = T,              LBOOK = T,
LTURB = F,              LDUMP = F,
&END,
```

TIME = 0.00 TIME-STEP = 0.00 STEP = 0

GRID	R	P	N	TE	TI	PTOTAL	BP	CONSTANT	BT
1	0.1250E 00	-0.3742E 04	0.3250E 16	0.9802E 00	0.9802E 00	0.1021E 05	0.5571E 06	0.5673E 06	-0.3716E 01
2	0.3750E 00	-0.3646E 04	0.8166E 16	0.2903E 01	0.2903E 01	0.3819E 05	0.5291E 06	0.5673E 06	-0.1895E 02
3	0.6250E 00	-0.3484E 04	0.5560E 16	0.4714E 01	0.4714E 01	0.8398E 05	0.4933E 06	0.5673E 06	-0.5684E 02
4	0.8750E 00	-0.3231E 04	0.7425E 16	0.6344E 01	0.6344E 01	0.1509E 06	0.4163E 06	0.5673E 06	-0.1345E 03
5	0.1125E 01	-0.2864E 04	0.9556E 16	0.7730E 01	0.7730E 01	0.2367E 06	0.3316E 06	0.5673E 06	-0.2719E 03
6	0.1375E 01	-0.2377E 04	0.1167E 17	0.8819E 01	0.8819E 01	0.3297E 06	0.2376E 06	0.5673E 06	-0.4832E 03
7	0.1625E 01	-0.1779E 04	0.1330E 17	0.9569E 01	0.9569E 01	0.4092E 06	0.1581E 06	0.5673E 06	-0.7666E 03
8	0.1875E 01	-0.1103E 04	0.1418E 17	0.9952E 01	0.9952E 01	0.4521E 06	0.1152E 06	0.5673E 06	-0.1096E 04
9	0.2125E 01	-0.3992E 03	0.1394E 17	0.9952E 01	0.9952E 01	0.4445E 06	0.1228E 06	0.5673E 06	-0.1433E 04
10	0.2375E 01	0.2765E 03	0.1270E 17	0.9569E 01	0.9569E 01	0.3896E 06	0.1777E 06	0.5673E 06	-0.1721E 04
11	0.2625E 01	0.3704E 03	0.1083E 17	0.8819E 01	0.8819E 01	0.3061E 06	0.2612E 06	0.5673E 06	-0.1925E 04
12	0.2875E 01	0.1362E 04	0.8750E 16	0.7730E 01	0.7730E 01	0.2178E 06	0.3495E 06	0.5673E 06	-0.2031E 04
13	0.3125E 01	0.1728E 04	0.7066E 16	0.6344E 01	0.6344E 01	0.1425E 06	0.4248E 06	0.5673E 06	-0.2047E 04
14	0.3375E 01	0.1982E 04	0.5796E 16	0.4714E 01	0.4714E 01	0.8756E 05	0.4797E 06	0.5673E 06	-0.2000E 04
15	0.3625E 01	0.2144E 04	0.5620E 16	0.2903E 01	0.2903E 01	0.5228E 05	0.5150E 06	0.5673E 06	-0.1915E 04
16	0.3875E 01	0.2239E 04	0.1015E 17	0.9802E 00	0.9802E 00	0.3198E 05	0.5354E 06	0.5673E 06	-0.1816E 04
17	0.4125E 01	0.2291E 04	0.1316E 17	0.5000E 00	0.5000E 00	0.2112E 05	0.5462E 06	0.5673E 06	-0.1715E 04
18	0.4375E 01	0.2317E 04	0.9916E 16	0.5000E 00	0.5000E 00	0.1539E 05	0.5519E 06	0.5673E 06	-0.1621E 04
19	0.4625E 01	0.2329E 04	0.8446E 16	0.5000E 00	0.5000E 00	0.1353E 05	0.5537E 06	0.5673E 06	-0.1534E 04
20	0.4875E 01	0.2334E 04	0.7631E 16	0.5000E 00	0.5000E 00	0.1255E 05	0.5547E 06	0.5673E 06	-0.1456E 04
21	0.5125E 01	0.2336E 04	0.7591E 16	0.5000E 00	0.5000E 00	0.1216E 05	0.5551E 06	0.5673E 06	-0.1385E 04
22	0.5375E 01	0.2336E 04	0.7503E 16	0.5000E 00	0.5000E 00	0.1202E 05	0.5553E 06	0.5673E 06	-0.1320E 04
23	0.5625E 01	0.2337E 04	0.7072E 16	0.5000E 00	0.5000E 00	0.1197E 05	0.5553E 06	0.5673E 06	-0.1261E 04
24	0.5875E 01	0.2337E 04	0.7461E 16	0.5000E 00	0.5000E 00	0.1195E 05	0.5553E 06	0.5673E 06	-0.1208E 04
25	0.6125E 01	0.2337E 04	0.7456E 16	0.5000E 00	0.5000E 00	0.1195E 05	0.5553E 06	0.5673E 06	-0.1158E 04
26	0.6375E 01	0.2337E 04	0.7453E 16	0.5000E 00	0.5000E 00	0.1194E 05	0.5553E 06	0.5673E 06	-0.1113E 04
27	0.6625E 01	0.2337E 04	0.7452E 16	0.5000E 00	0.5000E 00	0.1194E 05	0.5553E 06	0.5673E 06	-0.1071E 04
28	0.6875E 01	0.2308E 04	0.3754E 16	0.5000E 00	0.5000E 00	0.6015E 04	0.5613E 06	0.5673E 06	-0.1032E 04

GRID	R	JZ	JT	DELV	DELVZ	RESIST
1	0.1250E 00	-0.2168E 12	-0.7808E 12	0.5003E 06	0.1386E 06	0.1740E-12
2	0.3750E 00	-0.4103E 12	-0.1228E 13	0.6229E 06	0.2031E 06	0.3422E-13
3	0.6250E 00	-0.8059E 12	-0.1985E 13	0.7439E 06	0.3022E 06	0.1653E-13
4	0.8750E 00	-0.1434E 13	-0.2961E 13	0.8303E 06	0.4022E 06	0.1059E-13
5	0.1125E 01	-0.2291E 13	-0.4077E 13	0.8843E 06	0.4969E 06	0.7974E-14
6	0.1375E 01	-0.3232E 13	-0.5193E 13	0.9239E 06	0.5767E 06	0.6462E-14
7	0.1625E 01	-0.4078E 13	-0.6081E 13	0.9488E 06	0.6364E 06	0.5717E-14
8	0.1875E 01	-0.4579E 13	-0.6587E 13	0.9674E 06	0.6724E 06	0.5390E-14
9	0.2125E 01	-0.4572E 13	-0.6587E 13	0.9840E 06	0.6830E 06	0.5390E-14
10	0.2375E 01	-0.4061E 13	-0.6081E 13	0.9966E 06	0.6655E 06	0.5717E-14
11	0.2625E 01	-0.3296E 13	-0.5183E 13	0.9961E 06	0.6163E 06	0.6461E-14
12	0.2875E 01	-0.2250E 13	-0.4077E 13	0.9654E 06	0.5328E 06	0.7874E-14
13	0.3125E 01	-0.1403E 13	-0.2961E 13	0.8727E 06	0.4169E 06	0.1059E-13
14	0.3375E 01	-0.7766E 12	-0.1985E 13	0.7130E 06	0.2709E 06	0.1653E-13
15	0.3625E 01	-0.3809E 12	-0.1228E 13	0.4550E 06	0.1411E 06	0.3422E-13
16	0.3875E 01	-0.1648E 12	-0.7016E 12	0.1439E 06	0.3361E 05	0.1740E-12
17	0.4125E 01	-0.6229E 11	-0.3698E 12	0.5841E 05	0.9839E 03	0.4787E-12
18	0.4375E 01	-0.1987E 11	-0.1738E 12	0.3776E 05	0.4171E 04	0.4787E-12
19	0.4625E 01	-0.4662E 10	-0.8073E 11	0.1990E 05	0.1109E 04	0.4787E-12
20	0.4875E 01	-0.1530E 08	-0.3343E 11	0.8863E 04	0.4069E 01	0.4787E-12
21	0.5125E 01	0.1072E 10	-0.1277E 11	0.3562E 04	-0.2996E 03	0.4787E-12
22	0.5375E 01	0.1180E 10	-0.4495E 10	0.1247E 04	-0.3274E 03	0.4787E-12
23	0.5625E 01	0.1041E 10	-0.1460E 10	0.4085E 03	-0.2901E 03	0.4787E-12
24	0.5875E 01	0.8952E 09	-0.4381E 09	0.1223E 03	-0.2498E 03	0.4787E-12
25	0.6125E 01	0.7421E 09	-0.1196E 09	0.3160E 02	-0.2032E 03	0.4787E-12
26	0.6375E 01	0.6436E 09	-0.1632E [illegible]	0.4559E 01	-0.1798E 03	0.4787E-12
27	0.6625E 01	0.5743E 09	-0.1511E [illegible]	[illegible]	-0.1605E 03	0.4787E-12
28	0.6875E 01	0.3521E [illegible]	-0.1510E [illegible]	[illegible]	-0.1953E 04	0.4787E-12

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

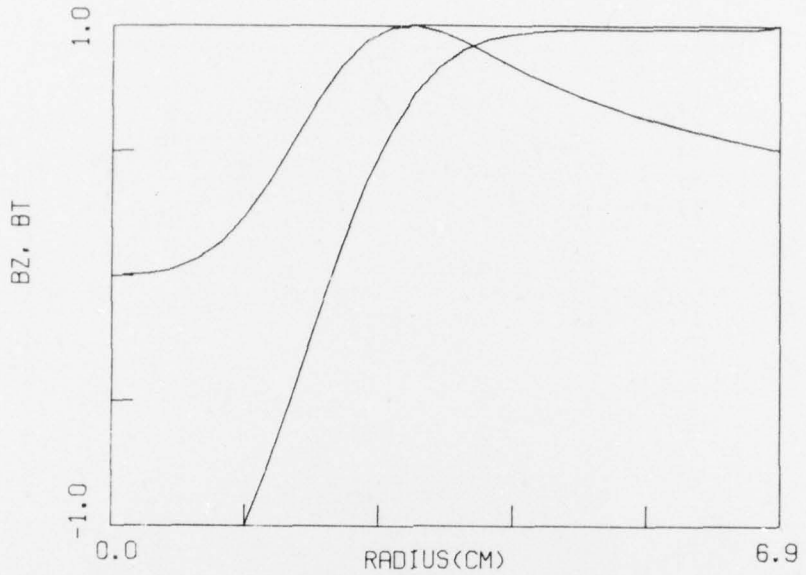
TIME = 3.40E-06 TIME-STEP = 1.00E-08 STEP = 340

GRID	R	B	N	TE	TI	PTOTAL	BP	CONSTANT	BT
1	0.1555E-01	-0.2062E 04	0.2099E 18	0.3010E 00	0.6125E 00	0.3072E 06	0.1696E 06	0.4768E 06	-0.5871E 02
2	0.7082E-01	-0.1946E 04	0.6842E 17	0.1101E 01	0.1959E 01	0.3244E 06	0.1524E 06	0.4768E 06	-0.1178E 03
3	0.1612E 00	-0.2259E 04	0.5318E 17	0.1068E 01	0.2102E 01	0.2718E 06	0.2050E 06	0.4768E 06	-0.4786E 02
4	0.2829E 00	-0.2098E 04	0.4042E 17	0.1587E 01	0.1924E 01	0.2274E 06	0.2494E 06	0.4768E 06	-0.1479E 03
5	0.4444E 00	-0.2524E 04	0.3342E 17	0.2126E 01	0.1871E 01	0.2140E 06	0.2628E 06	0.4768E 06	-0.3419E 03
6	0.6397E 00	-0.2221E 04	0.2989E 17	0.2721E 01	0.2502E 01	0.2501E 06	0.2268E 06	0.4768E 06	-0.6738E 03
7	0.8485E 00	-0.1606E 04	0.3072E 17	0.3114E 01	0.2813E 01	0.2917E 06	0.1851E 06	0.4768E 06	-0.1138E 04
8	0.1056E 01	-0.7486E 03	0.3043E 17	0.3275E 01	0.2952E 01	0.3036E 06	0.1732E 06	0.4768E 06	-0.1502E 04
9	0.1266E 01	0.1458E 03	0.2751E 17	0.3355E 01	0.3035E 01	0.2816E 06	0.1952E 06	0.4768E 06	-0.1577E 04
10	0.1483E 01	0.8032E 03	0.2271E 17	0.3421E 01	0.3107E 01	0.2396E 06	0.2372E 06	0.4768E 06	-0.1475E 04
11	0.1709E 01	0.1222E 04	0.1814E 17	0.3491E 01	0.3181E 01	0.1939E 06	0.2830E 06	0.4768E 06	-0.1377E 04
12	0.1939E 01	0.1484E 04	0.1414E 17	0.3560E 01	0.3253E 01	0.1544E 06	0.3225E 06	0.4768E 06	-0.1312E 04
13	0.2166E 01	0.1640E 04	0.1128E 17	0.3598E 01	0.3305E 01	0.1248E 06	0.3520E 06	0.4768E 06	-0.1267E 04
14	0.2386E 01	0.1729E 04	0.9450E 16	0.3579E 01	0.3318E 01	0.1044E 06	0.3724E 06	0.4768E 06	-0.1231E 04
15	0.2612E 01	0.1779E 04	0.8280E 16	0.3493E 01	0.3281E 01	0.8995E 05	0.3869E 06	0.4768E 06	-0.1198E 04
16	0.2957E 01	0.1818E 04	0.7327E 16	0.3276E 01	0.3139E 01	0.7530E 05	0.4015E 06	0.4768E 06	-0.1153E 04
17	0.3473E 01	0.1840E 04	0.6772E 16	0.2831E 01	0.2767E 01	0.6073E 05	0.4161E 06	0.4769E 06	-0.1100E 04
18	0.3975E 01	0.1863E 04	0.6397E 16	0.2365E 01	0.2332E 01	0.4814E 05	0.4287E 06	0.4768E 06	-0.1058E 04
19	0.4369E 01	0.1885E 04	0.6161E 16	0.1998E 01	0.1979E 01	0.3925E 05	0.4376E 06	0.4768E 06	-0.1027E 04
20	0.4718E 01	0.1904E 04	0.6041E 16	0.1683E 01	0.1671E 01	0.3246E 05	0.4444E 06	0.4769E 06	-0.9972E 03
21	0.5044E 01	0.1919E 04	0.6102E 16	0.1410E 01	0.1403E 01	0.2751E 05	0.4473E 06	0.4768E 06	-0.9655E 03
22	0.5348E 01	0.1928E 04	0.6442E 16	0.1190E 01	0.1187E 01	0.2453E 05	0.4523E 06	0.4768E 06	-0.9317E 03
23	0.5626E 01	0.1929E 04	0.7108E 16	0.1033E 01	0.1030E 01	0.2349E 05	0.4533E 06	0.4768E 06	-0.8976E 03
24	0.5873E 01	0.1923E 04	0.8015E 16	0.9363E 00	0.9242E 00	0.2389E 05	0.4529E 06	0.4768E 06	-0.8653E 03
25	0.6097E 01	0.1915E 04	0.8690E 16	0.9850E 00	0.8485E 00	0.2497E 05	0.4519E 06	0.4768E 06	-0.8359E 03
26	0.6358E 01	0.1910E 04	0.6126E 16	0.1852E 01	0.7225E 00	0.2526E 05	0.4516E 06	0.4768E 06	-0.8094E 03
27	0.6693E 01	0.1936E 04	0.5039E 16	0.2060E 01	0.4020E 00	0.1991E 05	0.4569E 06	0.4769E 06	-0.7888E 03
28	0.6938E 01	0.1940E 04	0.7496E 16	0.7926E 00	0.7926E 00	0.1904E 05	0.4578E 06	0.4768E 06	-0.7641E 03

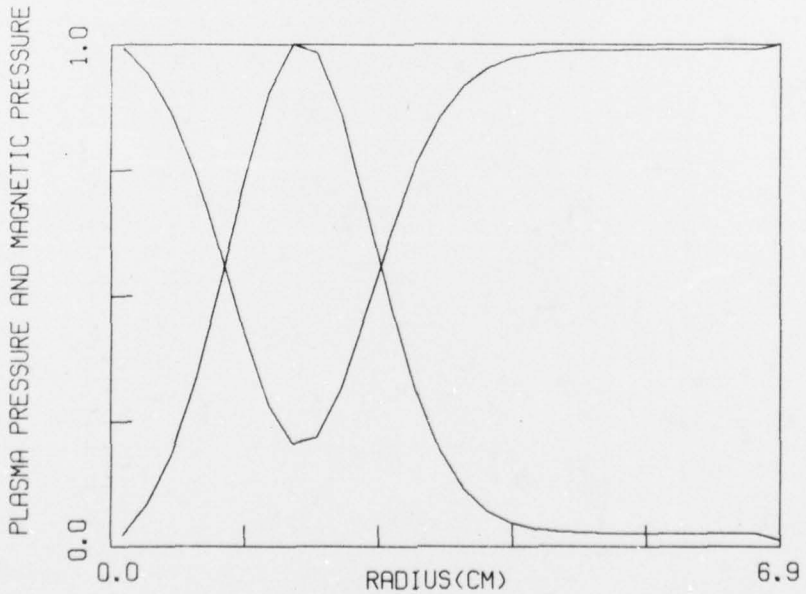
GRID	R	JZ	JT	DELV	DELVZ	RESIST
1	0.1555E-01	-0.1156E 14	-0.1029E 15	0.1020E 07	0.1147E 06	0.1025E-11
2	0.7082E-01	-0.3089E 13	0.1631E 13	-0.4962E 05	0.9400E 05	0.1066E-12
3	0.1612E 00	-0.1366E 13	0.6374E 13	-0.2496E 06	0.5348E 05	0.1071E-12
4	0.2829E 00	-0.3778E 13	0.2511E 13	-0.1273E 06	0.1946E 06	0.8461E-13
5	0.4444E 00	-0.5020E 13	-0.1584E 13	0.9864E 05	0.3376E 06	0.5461E-13
6	0.6397E 00	-0.7297E 13	-0.5370E 13	0.3741E 06	0.5084E 06	0.3771E-13
7	0.8485E 00	-0.7880E 13	-0.8453E 13	0.5729E 06	0.5341E 06	0.3079E-13
8	0.1056E 01	-0.5756E 13	-0.1002E 14	0.6858E 06	0.3938E 06	0.2855E-13
9	0.1266E 01	-0.2760E 13	-0.9698E 13	0.6584E 06	0.2089E 06	0.2753E-13
10	0.1483E 01	-0.1298E 13	-0.5827E 13	0.5236E 06	0.1180E 06	0.2675E-13
11	0.1709E 01	-0.1081E 13	-0.3574E 13	0.4103E 06	0.1241E 06	0.2594E-13
12	0.1939E 01	-0.1049E 13	-0.2179E 13	0.3248E 06	0.1544E 06	0.2519E-13
13	0.2166E 01	-0.9671E 12	-0.1301E 13	0.2400E 06	0.1788E 06	0.2079E-13
14	0.2386E 01	-0.8624E 12	-0.7465E 12	0.1645E 06	0.1900E 06	0.2500E-13
15	0.2612E 01	-0.7667E 12	-0.4411E 12	0.1048E 06	0.1926E 06	0.2592E-13
16	0.2957E 01	-0.6560E 12	-0.1854E 12	0.5268E 05	0.1864E 06	0.2854E-13
17	0.3473E 01	-0.5356E 12	-0.1058E 12	0.3253E 05	0.1647E 06	0.3550E-13
18	0.3975E 01	-0.4420E 12	-0.1197E 12	0.3876E 05	0.1440E 06	0.4659E-13
19	0.4369E 01	-0.3639E 12	-0.1313E 12	0.4437E 05	0.1230E 06	0.5993E-13
20	0.4718E 01	-0.2853E 12	-0.1210E 12	0.4185E 05	0.9835E 05	0.7753E-13
21	0.5044E 01	-0.2076E 12	-0.8866E 11	0.3032E 05	0.7084E 05	0.1011E-12
22	0.5348E 01	-0.1363E 12	-0.3730E 11	0.1207E 05	0.4404E 05	0.1303E-12
23	0.5626E 01	-0.7839E 11	0.2179E 11	-0.6382E 04	0.2295E 05	0.1612E-12
24	0.5873E 01	-0.3969E 11	0.6713E 11	-0.1796E 05	0.1030E 05	0.1868E-12
25	0.6097E 01	-0.4998E 11	0.6772E 11	-0.1623E 05	0.1198E 05	0.1842E-12
26	0.6358E 01	-0.1102E 12	-0.7064E 11	0.2401E 05	0.3747E 05	0.6717E-13
27	0.6693E 01	-0.8687E 11	-0.1147E 12	0.1739E 05	0.3556E 05	0.5706E-13
28	0.6938E 01	-0.1198E 12	-0.1227E 11	0.5519E 04	0.3328E 04	0.2398E-12

NTOTAL	TOTE	B-ENER	P-ENER	ITOTZ	ITOTT	PZTOTT	PTOTT	ETAM	BETAT
0.2543E 12	0.4041E-8	0.2712E 08	0.1219E-9	-0.7052E 10	-0.1297E 10	0.2627E 06	-0.7083E 04	0.1103E 01	[illegible]

0.7116E 00



(a)



(b)

Fig. 1 — Initial profiles of (a) magnetic field components B_z and B_θ and (b) plasma and magnetic pressure, taken from Ref. 7. All quantities are normalized to their peak values for display purposes.

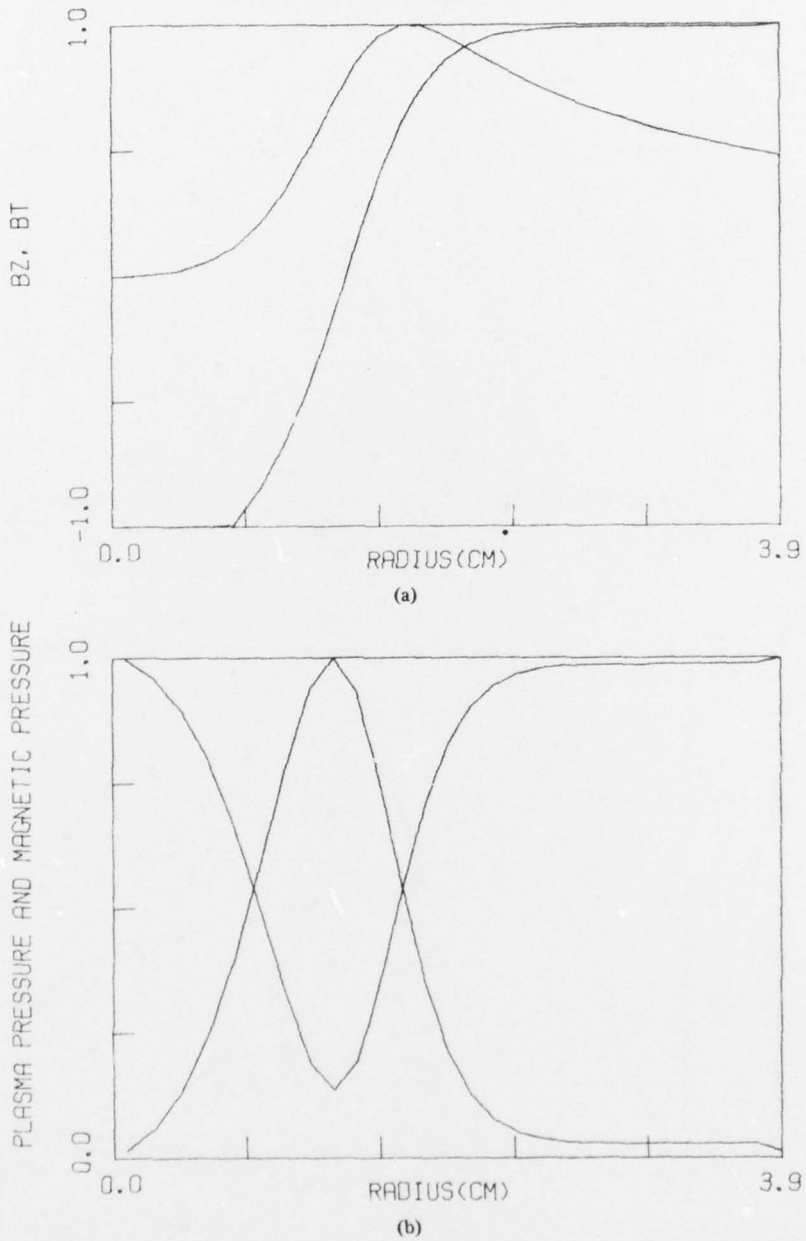


Fig. 2 — As in Fig. 1, after 0.3 msec of compression at 10^4 cm/sec, so that the outer boundary is at R_{wall} = 3.9 cm. All quantities are again normalized to their peak values.

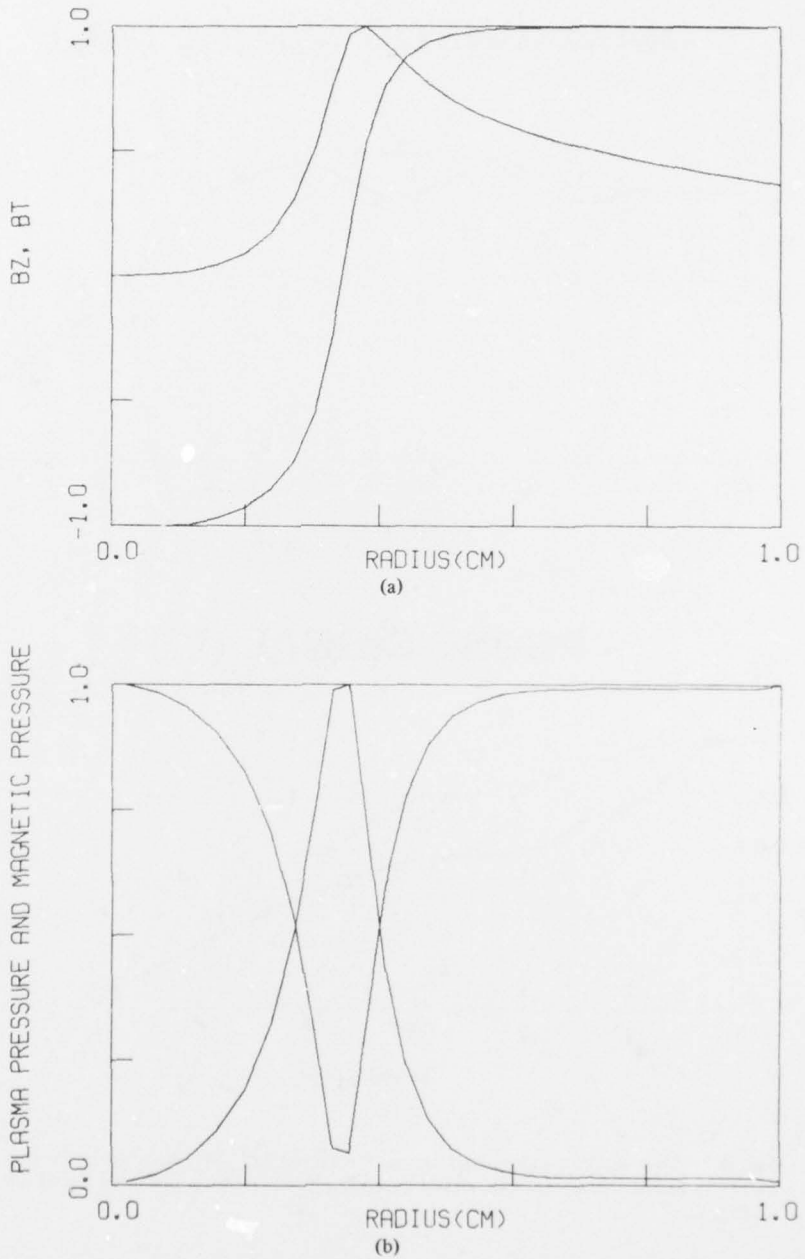
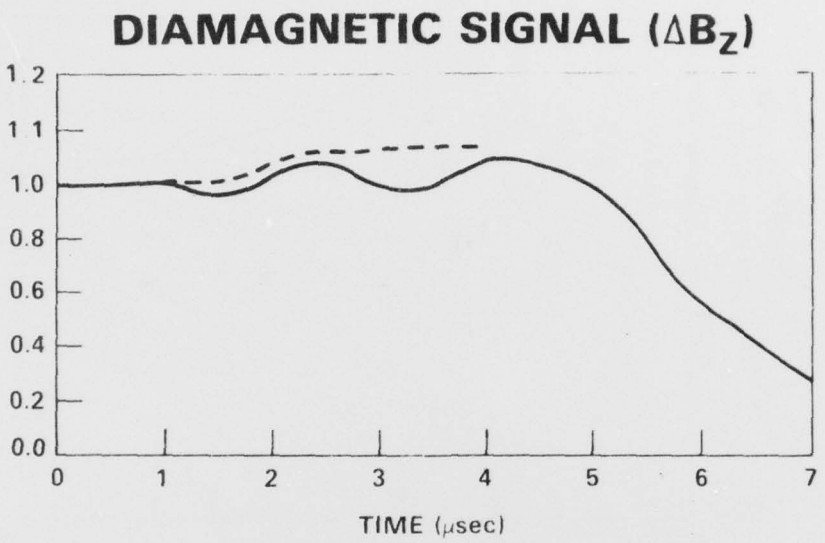
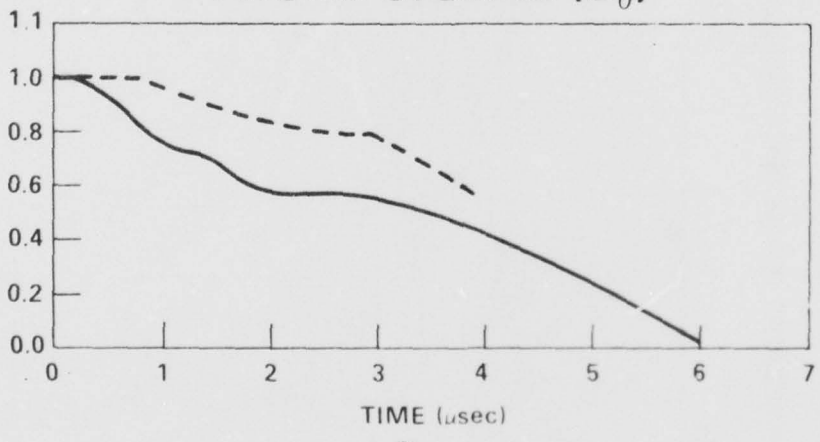


Fig. 3 — As in Figs. 1 and 2, after 0.6 msec of compression, with R_{wall} equal to about 15% of its initial value.



(a)



(b)

Fig. 4 — Diamagnetic signal and B_θ probe signal vs. time, as observed experimentally (solid trace) and calculated using QUEST, starting with the initial data of Fig. 1.

*U.S. GOVERNMENT PRINTING OFFICE: 1978 261-250/113 1-3